<https://gemini.google.com/u/1/app/2bce50b57d89d351>

<https://gemini.google.com/u/1/app/440838222ac514d7>

<https://gemini.google.com/u/1/app/1881f72a50cc0fbc>

\* \* \* \* \*

User & Content Infrastructure Module Overview

=============================================

Version: 2.3 (Clarifications on Translatable Field Handling and API Representation)

Date: May 18, 2025

This document provides a master recap and architecture overview for the User & Content Infrastructure Module of the pilgrimage-platform database. It details how tables, ENUMs, translation mechanisms, security policies, API considerations, and other database objects interlock, along with a recommended build order for successful deployment. The five key tables forming this module are `profiles`, `user\_roles\_master`, `languages\_master`, `media`, and `translations`.

1\. Executive Summary

---------------------

This database module establishes the foundational framework for managing users, their roles (synchronized to auth.users.raw\_app\_meta\_data.roles for JWT inclusion), multilingual content, and media assets within the pilgrimage platform. It enables core functionalities such as user authentication extensions (via Supabase Auth), role-based access control (RBAC), personalized user experiences, user activity tracking (via profiles.last\_activity\_at), and a centralized system for handling all platform media.

Key enhancements include standardized audit columns (created\_by\_profile\_id, updated\_by\_profile\_id) across all tables for better tracking, and icon\_identifier fields for improved UI consistency in master tables.

A core principle for multilingual content is that main table columns designated as translatable (e.g., user\_roles\_master.default\_display\_name, media.default\_alt\_text) store their content directly in the primary reference language (English). The public.translations table then stores the corresponding text for all other supported languages. The API representation makes this clear by providing the base field (e.g., default\_alt\_text containing English) and a corresponding default\_alt\_text\_translations object for other languages.

Primary business goals unlocked include:

- Secure and distinct user data management extending Supabase's authentication.

- Flexible permission systems through defined user roles available in JWTs.

- Global reach via robust multilingual support for platform content and UI, with English as the primary reference language directly in main tables and other languages managed through a central `translations` table.

- Efficient management and delivery of media assets like images, with comprehensive audit trails and translatable descriptive text.

\* \* \* \* \*

2\. Group-Level Snapshot

------------------------

| Group | Key Tables | Primary Purpose | Top Inter-Group Links |

| 1\. User & Content Infrastructure Module | `profiles`, `user\_roles\_master`, `languages\_master`, `media`, `translations` | Manages user-specific data (including roles synchronized to auth system), defines roles, lists supported languages, stores media metadata, and holds translations for content across the platform. Primary language (English) content resides in main tables; other languages are in `translations`. | `profiles` to `auth.users` (Supabase built-in, including `raw\_app\_meta\_data.roles` sync) &lt;br> `profiles` to `media` (for avatars) &lt;br> `profiles` to `languages\_master` (for preferred language) &lt;br> All tables to `profiles` (for `created\_by\_profile\_id`, `updated\_by\_profile\_id` audit fields) &lt;br> Various tables to `translations` & `media`. |

\* \* \* \* \*

3\. Narrative Walkthrough

-------------------------

This module forms the bedrock for user identity, permissions, internationalization (i18n), and media management.

- `languages\_master` (v2.1):

- Definitive list of supported languages for the platform.

- Stores `language\_code`, native name (`display\_name\_native`), English name (`display\_name\_en` - this specific `\_en` suffix is intentional in the DB), `icon\_identifier`, active/primary flags, and standard audit columns.

- Powers language selectors and guides the `translations` system. `is\_primary\_content\_language` flags the reference language (English).

- `user\_roles\_master` (v2.1):

- Defines all application roles (e.g., 'pilgrim\_user').

- Holds `role\_code`, the default display name and description in the primary reference language (English) directly in `default\_display\_name` and `default\_description` columns, `icon\_identifier`, hierarchy level, system/active flags, and standard audit columns. These English texts are translatable via `public.translations` for other languages.

- Source for role definitions. `AFTER DELETE` trigger cleans associated entries in `public.translations`. `profiles.roles` are validated against this table.

- `profiles` (v2.3):

- Extends `auth.users` (1:1). Stores app-specific user data.

- Includes `roles` array (source of truth, synchronized to `auth.users.raw\_app\_meta\_data.roles` via triggers), preferences, public info (e.g., `public\_bio` stored in language of submission or primary reference language), `account\_status`, `last\_activity\_at`, and standard audit columns. `public\_bio` can be translated via `public.translations` if it becomes centrally managed.

- Triggers: `handle\_new\_user` (creates profile, sets initial roles in `profiles` and `auth.users`), `check\_profile\_roles` (validates roles), `sync\_profile\_roles\_to\_auth\_user` (updates `auth.users.raw\_app\_meta\_data.roles` on `profiles.roles` change).

- `media` (v2.2):

- Central metadata for media files.

- Stores Storage refs, `uploader\_profile\_id` (effectively `created\_by\_profile\_id`), `updated\_by\_profile\_id` (for moderation/admin edits), default alternative text (`default\_alt\_text`) and caption (`default\_caption`) in the primary reference language (English) directly in their respective columns, `image\_variants\_json`, `media\_status`, `last\_linked\_or\_used\_at`. These English texts are translatable via `public.translations`.

- `AFTER DELETE` trigger cleans associated entries in `public.translations`. ENUMs for types, licenses, statuses.

- `translations` (v2.1):

- Stores translated text for fields from other tables that are not in the primary reference language (English).

- Links `language\_code` to text identified by `table\_identifier`, `column\_identifier`, `row\_foreign\_key`.

- Includes `translation\_status`, `translated\_by\_profile\_id` (content author), `translation\_last\_updated\_at`, and standard row audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`).

- Orphaned translation cleanup is critical and handled by `AFTER DELETE` triggers on the parent tables (e.g., `user\_roles\_master`, `media`).

\* \* \* \* \*

4\. Cross-Cutting Concerns

--------------------------

- Users & Roles:

- `profiles.roles` is the primary store for a user's roles, synchronized to `auth.users.raw\_app\_meta\_data.roles` for JWT inclusion.

- Standard audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id` referencing `profiles.id`) are on all tables in this module. `uploader\_profile\_id` in `media` serves as its `created\_by\_profile\_id`.

- Translations / i18n:

- `languages\_master` defines supported languages. The primary reference language (English) content is stored directly in the main columns of translatable tables (e.g., `user\_roles\_master.default\_display\_name`, `media.default\_alt\_text`). The only exception is `languages\_master.display\_name\_en`, which is an explicit column for the English name of a language.

- The `public.translations` table stores text for all other languages.

- API representation will provide the base field containing the English text (e.g., `default\_alt\_text`) and a corresponding `default\_alt\_text\_translations` object (e.g. `{ "it": "Testo alternativo italiano" }`) for other languages. If a `lang` parameter is used in an API request, the base field itself may be resolved to the requested language.

- `AFTER DELETE` triggers on parent tables (e.g., `user\_roles\_master`, `media`) are critical for removing orphaned `translations` entries.

- ENUM & Taxonomy Registry:

- `profiles`: `units\_preference\_enum`, `pilgrim\_experience\_enum`, `user\_account\_status\_enum`.

- `media`: `media\_asset\_type\_enum`, `media\_licence\_enum`, `media\_status\_enum`.

- `languages\_master`, `user\_roles\_master` include `icon\_identifier`.

- Media & Files:

- Files reside in Supabase Storage.

- The `media` table holds metadata, including `image\_variants\_json` (populated by application/backend processes) and `last\_linked\_or\_used\_at` to aid in cleaning up unlinked media.

- Audit / Soft-Delete / Versioning:

- Standard Audit Columns: All tables include `created\_at`, `updated\_at` (with `extensions.moddatetime` trigger or equivalent), `created\_by\_profile\_id`, and `updated\_by\_profile\_id`. For `media`, `uploader\_profile\_id` is analogous to `created\_by\_profile\_id`.

- `profiles` includes `last\_activity\_at`.

- Soft Deletes: `profiles.account\_status` manages user account lifecycle; `user\_roles\_master.deleted\_at` (complemented by `is\_role\_active`); `media.deleted\_at`.

\* \* \* \* \*

5\. Security & Access Control 🔐

--------------------------------

- Authentication Provider: Supabase Auth is utilized, leveraging JWTs.

- Roles from `public.profiles.roles` are synchronized to `auth.users.raw\_app\_meta\_data.roles` for availability in JWT claims.

- RLS Overview: Extensively applied across all tables. Policies are defined in individual table specifications.

- Dedicated SECURITY DEFINER Functions:

- `public.check\_profile\_roles()`

- `public.handle\_new\_user()` (updated to also set `auth.users.raw\_app\_meta\_data.roles` and `profiles.last\_activity\_at`)

- `public.sync\_profile\_roles\_to\_auth\_user()` (new, for `profiles` trigger to update `auth.users`)

- `public.cleanup\_user\_roles\_master\_translations()`

- `public.cleanup\_media\_translations()`

- `public.cleanup\_profile\_translations()` (conditional, if `profiles.public\_bio` becomes system-translatable)

- Conceptual `public.update\_media\_last\_linked\_timestamp()` to be triggered by media linking tables.

- Key Helper Functions for RLS (typically `STABLE SECURITY INVOKER` or `SECURITY DEFINER` with caution):

- `public.has\_role(TEXT)` (checks current authenticated user's roles from `profiles` or JWT claim)

- `public.has\_role\_on\_profile(UUID, TEXT)` (checks a specific user's roles)

- (Other administrative check functions like `is\_platform\_admin()` can be built using these.)

\* \* \* \* \*

6\. API Endpoints Summary

-------------------------

This module provides foundational data for users, content structure, and internationalization. Key conceptual API endpoints, as detailed in the "API Specification Conceptualization - User & Content Infrastructure Module (v1.0)", include:

- `GET /profiles/me`: Retrieves the current authenticated user's profile. Translatable fields like `public\_bio` will be returned in the language specified by the `lang` parameter if a translation exists, otherwise in the primary reference language (English). A `public\_bio\_translations` object may also provide all available language versions. Role details and avatar information can be included.

- `GET /languages`: Lists available platform languages, filterable and sortable.

- `GET /media/{media\_id}`: Fetches a media item's details. Translatable fields like `default\_alt\_text` and `default\_caption` are resolved based on the `lang` parameter, with the base field containing English and a `\*\_translations` object providing other versions.

The API design ensures that the primary reference language (English) content is directly available in base fields (e.g., `media.default\_alt\_text`), with translations for other languages accessible via a corresponding `[fieldname]\_translations` object.

\* \* \* \* \*

7\. Prerequisite Objects & Build Order ⚙️

-----------------------------------------

The build order assumes `auth.users` table and `extensions.moddatetime` function (or an equivalent `handle\_updated\_at` that also manages `updated\_by\_profile\_id`) exist. Foreign keys for audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`) on `languages\_master`, `user\_roles\_master`, and `media` linking to `public.profiles` are typically added via `ALTER TABLE` after `public.profiles` is created to manage dependencies.

1. Global Helper Functions (SQL/PLPGSQL)

(Define public.has\_role(TEXT) and public.has\_role\_on\_profile(UUID, TEXT). Their full implementation depends on profiles.)

SQL

```

-- Example stubs if profiles not yet available for full function definition

CREATE OR REPLACE FUNCTION public.has\_role(role\_to\_check TEXT) RETURNS BOOLEAN AS $$BEGIN RETURN FALSE; END;$$ LANGUAGE plpgsql STABLE SECURITY INVOKER; -- [cite: 209]

CREATE OR REPLACE FUNCTION public.has\_role\_on\_profile(target\_user\_id uuid, role\_to\_check TEXT) RETURNS BOOLEAN AS $$BEGIN RETURN FALSE; END;$$ LANGUAGE plpgsql STABLE SECURITY INVOKER; -- [cite: 210]

```

2. Types & ENUMs

SQL

```

DO $$ BEGIN

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'units\_preference\_enum') THEN CREATE TYPE public.units\_preference\_enum AS ENUM ('metric', 'imperial'); END IF; -- [cite: 211, 483]

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'pilgrim\_experience\_enum') THEN CREATE TYPE public.pilgrim\_experience\_enum AS ENUM ('novice\_first\_pilgrimage', 'intermediate\_few\_pilgrimages', 'experienced\_many\_pilgrimages', 'long\_distance\_veteran'); END IF; -- [cite: 212, 483]

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'user\_account\_status\_enum') THEN CREATE TYPE public.user\_account\_status\_enum AS ENUM ('active', 'pending\_verification', 'email\_unconfirmed', 'suspended\_by\_admin', 'deactivated\_by\_user'); END IF; -- [cite: 213, 484]

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_asset\_type\_enum') THEN CREATE TYPE public.media\_asset\_type\_enum AS ENUM ('image', 'document\_pdf', 'audio\_clip', 'video\_clip', 'gpx\_file', 'other\_file'); END IF; -- [cite: 214, 1101]

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_licence\_enum') THEN CREATE TYPE public.media\_licence\_enum AS ENUM ('all\_rights\_reserved', 'cc\_by', 'cc\_by\_sa', 'cc\_by\_nd', 'cc\_by\_nc', 'cc\_by\_nc\_sa', 'cc\_by\_nc\_nd', 'cc0\_public\_domain', 'uploader\_owns\_contact\_for\_use', 'official\_permission\_granted', 'unknown\_licence'); END IF; -- [cite: 215, 1102]

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_status\_enum') THEN CREATE TYPE public.media\_status\_enum AS ENUM ('processing\_upload', 'pending\_review', 'published\_approved', 'rejected\_hidden', 'archived', 'error\_uploading'); END IF; -- [cite: 216, 1103]

END$$;

```

3. Core Tables & Deferred FK Constraints

(DDL extracts based on individual table specs V2.1/V2.2/V2.3)

SQL

```

-- languages\_master table (v2.1)

CREATE TABLE public.languages\_master ( -- [cite: 218]

language\_code text NOT NULL PRIMARY KEY,

display\_name\_native text NOT NULL, -- [cite: 770]

display\_name\_en text NOT NULL, -- [cite: 775] Note: "\_en" is intentional here

icon\_identifier text NULL, -- [cite: 779]

is\_active\_for\_platform boolean NOT NULL DEFAULT false, -- [cite: 783]

is\_primary\_content\_language boolean NOT NULL DEFAULT false, -- [cite: 786]

display\_order\_ui integer NULL UNIQUE, -- [cite: 790]

created\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 218, 793]

updated\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 218, 796]

created\_by\_profile\_id uuid NULL, -- [cite: 218, 800]

updated\_by\_profile\_id uuid NULL, -- [cite: 218, 804]

CONSTRAINT check\_language\_code\_format CHECK (language\_code ~ '^[a-z]{2}(-[A-Z]{2})?$' AND char\_length(language\_code) >= 2 AND char\_length(language\_code) <= 5) -- [cite: 218, 765]

);

-- user\_roles\_master table (v2.1)

CREATE TABLE public.user\_roles\_master ( -- [cite: 219]

role\_code text NOT NULL PRIMARY KEY, -- [cite: 595]

default\_display\_name text NOT NULL, -- Stores English [cite: 219, 601]

default\_description text NULL, -- Stores English [cite: 219, 607]

icon\_identifier text NULL, -- [cite: 219, 613]

permissions\_summary\_json jsonb NULL, -- [cite: 220, 617]

role\_hierarchy\_level integer NULL, -- [cite: 220, 623]

is\_system\_role boolean NOT NULL DEFAULT false, -- [cite: 220, 629]

is\_role\_active boolean NOT NULL DEFAULT true, -- [cite: 220, 635]

default\_for\_new\_pilgrim\_users boolean NOT NULL DEFAULT false, -- [cite: 220, 639]

created\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 220, 644]

updated\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 220, 647]

created\_by\_profile\_id uuid NULL, -- [cite: 220, 651]

updated\_by\_profile\_id uuid NULL, -- [cite: 220, 655]

deleted\_at timestamp with time zone NULL, -- [cite: 220, 659]

CONSTRAINT check\_role\_code\_format CHECK (role\_code = lower(role\_code) AND role\_code ~ '^[a-z0-9\_]+$') -- [cite: 220, 594]

);

-- media table (v2.2)

CREATE TABLE public.media ( -- [cite: 221]

id uuid NOT NULL DEFAULT gen\_random\_uuid() PRIMARY KEY, -- [cite: 1012]

uploader\_profile\_id uuid NULL, -- Serves as created\_by [cite: 221, 1015]

storage\_bucket\_name text NOT NULL, -- [cite: 221, 1020]

storage\_object\_path\_original text NOT NULL UNIQUE, -- [cite: 221, 1023]

file\_name\_original text NOT NULL, -- [cite: 221, 1026]

file\_mime\_type text NOT NULL, -- [cite: 221, 1030]

file\_size\_bytes\_original bigint NULL CHECK (file\_size\_bytes\_original IS NULL OR file\_size\_bytes\_original > 0), -- [cite: 221, 1033]

image\_width\_px\_original integer NULL CHECK (image\_width\_px\_original IS NULL OR image\_width\_px\_original > 0), -- [cite: 222, 1036]

image\_height\_px\_original integer NULL CHECK (image\_height\_px\_original IS NULL OR image\_height\_px\_original > 0), -- [cite: 222, 1039]

image\_variants\_json jsonb NULL, -- [cite: 222, 1044]

default\_alt\_text text NULL, -- Stores English [cite: 222, 1048]

default\_caption text NULL, -- Stores English [cite: 222, 1052]

attribution\_text text NULL, -- [cite: 222, 1056]

attribution\_url text NULL CHECK (attribution\_url IS NULL OR attribution\_url ~\* '^https?://.+'), -- [cite: 222, 1062]

licence public.media\_licence\_enum NULL DEFAULT 'all\_rights\_reserved', -- [cite: 222, 1065]

media\_asset\_type public.media\_asset\_type\_enum NOT NULL DEFAULT 'image', -- [cite: 222, 1067]

media\_status public.media\_status\_enum NOT NULL DEFAULT 'pending\_review', -- [cite: 222, 1071]

tags text[] NULL, -- [cite: 223, 1074]

checksum\_sha256\_original text NULL UNIQUE, -- [cite: 223, 1077]

dominant\_color\_hex text NULL CHECK (dominant\_color\_hex IS NULL OR dominant\_color\_hex ~ '^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$'), -- [cite: 223, 1081]

last\_linked\_or\_used\_at timestamp with time zone NULL, -- [cite: 223, 1085]

created\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 223, 1088]

updated\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 223, 1090]

updated\_by\_profile\_id uuid NULL, -- [cite: 224, 1095]

deleted\_at timestamp with time zone NULL -- [cite: 224, 1098]

);

-- profiles table (v2.3)

CREATE TABLE public.profiles ( -- [cite: 224]

id uuid NOT NULL PRIMARY KEY REFERENCES auth.users(id) ON DELETE CASCADE, -- [cite: 224, 381]

roles text[] NOT NULL, -- [cite: 224, 385]

username text NULL UNIQUE, -- [cite: 224, 390]

full\_name text NULL, -- [cite: 224, 394]

public\_display\_name text NULL UNIQUE, -- [cite: 224, 399]

public\_avatar\_media\_id uuid NULL REFERENCES public.media(id) ON DELETE SET NULL, -- [cite: 225, 403]

public\_bio text NULL, -- Stores content in primary ref lang or lang of submission [cite: 225, 407]

preferred\_language\_code text NOT NULL DEFAULT 'en' REFERENCES public.languages\_master(language\_code) ON UPDATE CASCADE ON DELETE RESTRICT, -- [cite: 225, 412]

preferred\_units\_of\_measure public.units\_preference\_enum NOT NULL DEFAULT 'metric', -- [cite: 225, 415]

preferred\_timezone text NULL DEFAULT 'Europe/Rome', -- [cite: 225, 419]

pilgrim\_experience\_level public.pilgrim\_experience\_enum NULL, -- [cite: 226, 422]

pilgrimage\_interests\_tags text[] NULL, -- [cite: 226, 426]

contributor\_organization\_name text NULL, -- [cite: 226, 430]

contributor\_organization\_role text NULL, -- [cite: 226, 434]

contact\_public\_email text NULL, -- [cite: 226, 438]

website\_url\_profile text NULL, -- [cite: 226, 442]

notification\_preferences\_json jsonb NULL, -- [cite: 226, 445]

is\_profile\_publicly\_visible boolean NOT NULL DEFAULT false, -- [cite: 226, 449]

contribution\_score integer NOT NULL DEFAULT 0, -- [cite: 226, 453]

account\_status public.user\_account\_status\_enum NOT NULL DEFAULT 'active', -- [cite: 226, 456]

terms\_accepted\_at timestamp with time zone NULL, -- [cite: 226, 460]

last\_login\_at timestamp with time zone NULL, -- [cite: 226, 463]

last\_activity\_at timestamp with time zone NULL, -- [cite: 226, 467]

created\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 226, 471]

updated\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 227, 475]

updated\_by\_profile\_id uuid NULL REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 227, 480]

CONSTRAINT check\_preferred\_language\_code\_format CHECK (preferred\_language\_code ~ '^[a-z]{2}(-[A-Z]{2})?$'), -- [cite: 227, 412]

CONSTRAINT check\_contact\_public\_email\_format CHECK (contact\_public\_email IS NULL OR contact\_public\_email ~\* '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'), -- [cite: 227, 438]

CONSTRAINT check\_website\_url\_profile\_format CHECK (website\_url\_profile IS NULL OR website\_url\_profile ~\* '^https?://.+'), -- [cite: 227, 442]

CONSTRAINT check\_contribution\_score\_non\_negative CHECK (contribution\_score >= 0) -- [cite: 227, 454]

);

-- Add Deferred Foreign Key Constraints for audit columns

ALTER TABLE public.languages\_master ADD CONSTRAINT fk\_languages\_created\_by FOREIGN KEY (created\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 227]

ALTER TABLE public.languages\_master ADD CONSTRAINT fk\_languages\_updated\_by FOREIGN KEY (updated\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 228]

ALTER TABLE public.user\_roles\_master ADD CONSTRAINT fk\_user\_roles\_created\_by FOREIGN KEY (created\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 229]

ALTER TABLE public.user\_roles\_master ADD CONSTRAINT fk\_user\_roles\_updated\_by FOREIGN KEY (updated\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 230]

ALTER TABLE public.media ADD CONSTRAINT fk\_media\_uploader\_profile FOREIGN KEY (uploader\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 231]

ALTER TABLE public.media ADD CONSTRAINT fk\_media\_updated\_by\_profile FOREIGN KEY (updated\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL; -- [cite: 232]

-- translations table (v2.1)

CREATE TABLE public.translations ( -- [cite: 235]

id bigserial PRIMARY KEY, -- [cite: 235, 910]

table\_identifier text NOT NULL CHECK (char\_length(table\_identifier) > 0), -- [cite: 235, 913]

column\_identifier text NOT NULL CHECK (char\_length(column\_identifier) > 0), -- [cite: 235, 917]

row\_foreign\_key text NOT NULL, -- [cite: 235, 921] PK from source table as text

language\_code text NOT NULL REFERENCES public.languages\_master(language\_code) ON UPDATE CASCADE ON DELETE RESTRICT, -- [cite: 235, 926]

translated\_text text NOT NULL, -- [cite: 235, 929]

translation\_status text NULL CHECK (translation\_status IN ('draft', 'needs\_review', 'published\_live', 'needs\_update') OR translation\_status IS NULL), -- [cite: 235, 933]

translated\_by\_profile\_id uuid NULL REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 235, 935]

translation\_last\_updated\_at timestamp with time zone NULL, -- [cite: 236, 938]

created\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 236, 941]

updated\_at timestamp with time zone NOT NULL DEFAULT now(), -- [cite: 236, 944]

created\_by\_profile\_id uuid NULL REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 236, 948]

updated\_by\_profile\_id uuid NULL REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 236, 952]

CONSTRAINT uq\_translation\_entry UNIQUE (table\_identifier, column\_identifier, row\_foreign\_key, language\_code) -- [cite: 236, 955]

);

```

4. Trigger Functions

(These include check\_profile\_roles, handle\_new\_user (updated for role sync and last\_activity\_at), sync\_profile\_roles\_to\_auth\_user, cleanup\_user\_roles\_master\_translations, cleanup\_media\_translations, and the conceptual update\_media\_last\_linked\_timestamp.)

The definitions for these functions as provided in profiles.docx (v2.3 sections 4) and other specifications (e.g., media spec for its translation cleanup ) are assumed.

5. Views

SQL

```

CREATE OR REPLACE VIEW public.public\_profiles\_view AS

SELECT p.id, p.public\_display\_name, p.public\_avatar\_media\_id, m.storage\_object\_path\_original AS public\_avatar\_url, -- Example, consider using image\_variants\_json for actual URL

p.public\_bio, p.pilgrimage\_interests\_tags, p.contributor\_organization\_name, p.website\_url\_profile

FROM public.profiles p LEFT JOIN public.media m ON p.public\_avatar\_media\_id = m.id

WHERE p.is\_profile\_publicly\_visible = true AND p.account\_status = 'active'::public.user\_account\_status\_enum; -- [cite: 237]

```

6. Indexes & Constraints (Additional)

(Indexes are consolidated from individual table specs. For example, GIN indexes on profiles.roles and media.tags, and B-tree indexes on status fields and audit FKs like idx\_profiles\_last\_activity\_at are important).

7. Triggers (Application of DML Triggers)

(Apply all updated\_at triggers, role validation/sync triggers (check\_profile\_roles, sync\_profile\_roles\_to\_auth\_user on profiles), the new user trigger (handle\_new\_user on auth.users), and translation cleanup triggers (e.g., on user\_roles\_master, media)).

8. RLS Policies

(Enable RLS on all tables. Policy definitions detailed in individual table specs should be applied consistently, using helper functions like public.has\_role()).

9. Seed Data:

- Seed data for `languages\_master` (v2.1 Section 13) and `user\_roles\_master` (v2.1 Section 13) should be populated, ensuring new columns like `icon\_identifier` and audit columns are included with appropriate placeholders like `[ADMIN\_UUID]`.

\* \* \* \* \*

8\. Performance & Optimization Extras

-------------------------------------

- Key Indexes: Comprehensive indexing, as defined in individual table specifications (versions 2.1, 2.2, 2.3), including for audit columns and `profiles.last\_activity\_at`, is crucial.

- Database Functions for Complex API Responses: As emphasized, creating PostgreSQL functions to assemble complex JSON responses involving joins and translation resolution (e.g., for `GET /profiles/me`) is highly recommended to optimize performance and simplify API gateway logic.

- Full-Text Search: Noted as a V2+ improvement for text fields in `media`, `profiles` (`public\_bio`), and `translations` (`translated\_text`).

- Partitioning: `media` and `translations` tables are candidates for partitioning post-V2 if they grow very large.

- Caching: `languages\_master` and `user\_roles\_master` are excellent candidates for application-level caching due to their relatively static nature and frequent access.

\* \* \* \* \*

9\. Visuals (Conceptual ERD - Key Tables & Links)

-------------------------------------------------

\*(The Mermaid ERD provided in the original document v2.2, Section 9, generally remains valid for depicting key relationships within this module, reflecting audit links and translation connections. Key is that `user\_roles\_master.default\_display\_name` (English) is in the table, and `translations` links for other languages.)\*

Code snippet

```

erDiagram

auth\_users ||--o{ profiles : "has\_profile"

profiles }o--|| languages\_master : "prefers\_language"

profiles }o--|| media : "has\_avatar (optional)"

profiles }o--o| profiles : "updated\_by (admin)"

profiles {

uuid id PK

text[] roles

text public\_bio # Stores English or submission lang

timestamptz last\_activity\_at

uuid updated\_by\_profile\_id FK

uuid created\_by\_profile\_id FK "Implicitly self or system"

}

profiles ..> translations : "public\_bio\_translated\_in (conceptual)"

user\_roles\_master {

text role\_code PK

text default\_display\_name # Stores English

text icon\_identifier

uuid created\_by\_profile\_id FK

uuid updated\_by\_profile\_id FK

}

profiles ..> user\_roles\_master : "has\_roles"

user\_roles\_master }o--o| profiles : "audit\_user\_roles\_created"

user\_roles\_master }o--o| profiles : "audit\_user\_roles\_updated"

user\_roles\_master ..> translations : "default\_display\_name\_translated\_in"

languages\_master {

text language\_code PK

text display\_name\_en # Explicit English name column

text icon\_identifier

uuid created\_by\_profile\_id FK

uuid updated\_by\_profile\_id FK

}

languages\_master }o--o| profiles : "audit\_languages\_created"

languages\_master }o--o| profiles : "audit\_languages\_updated"

media {

uuid id PK

text default\_alt\_text # Stores English

uuid uploader\_profile\_id FK "is\_created\_by"

uuid updated\_by\_profile\_id FK "is\_updated\_by\_admin\_moderator"

}

media }o--|| profiles : "uploaded\_by"

media }o--o| profiles : "updated\_by\_admin"

media ..> translations : "default\_alt\_text\_translated\_in"

translations {

bigint id PK

text table\_identifier

text column\_identifier

text row\_foreign\_key

text language\_code FK

text translated\_text

uuid translated\_by\_profile\_id FK

uuid created\_by\_profile\_id FK

uuid updated\_by\_profile\_id FK

}

translations }o--|| languages\_master : "is\_in\_language"

translations }o--o| profiles : "content\_translated\_by"

translations }o--o| profiles : "row\_created\_by"

translations }o--o| profiles : "row\_updated\_by"

```

\* \* \* \* \*

10\. Critical Gaps & Risks

--------------------------

- 🔴 Implementation of RLS Helper Functions: Secure and correct implementation of `public.has\_role(TEXT)`, `public.has\_role\_on\_profile(UUID, TEXT)` remains paramount for effective access control.

- 🔴 Orphaned Translations Cleanup Triggers: These `AFTER DELETE` triggers \*must\* be implemented correctly on ALL parent tables with translatable fields (e.g., `user\_roles\_master`, `media`, and potentially `profiles` if `public\_bio` becomes system-translatable) to maintain data integrity in `public.translations`.

- 🟠 Population Logic for Audit Fields & `last\_activity\_at`:

- The application layer or specific database triggers need to correctly populate `created\_by\_profile\_id` and `updated\_by\_profile\_id` (especially for admin actions or system processes where `auth.uid()` from a JWT might not be the direct actor, or for setting these on behalf of another user).

- The mechanism for updating `profiles.last\_activity\_at` (e.g., via API calls on significant user actions, or specific DB triggers on related tables like `user\_waypoint\_short\_tips`, `accommodation\_reviews`) needs to be robustly defined and implemented.

- 🟠 Role Synchronization Robustness: Thoroughly test the `profiles.roles` to `auth.users.raw\_app\_meta\_data.roles` synchronization handled by the `handle\_new\_user` and `sync\_profile\_roles\_to\_auth\_user` triggers.

- 🟠 Backend Media Processing: The full pipeline for media (variant generation for `image\_variants\_json`, checksum calculation, dominant color extraction, etc.) is a significant backend development task.

- 🟢 Clarity on `lang` Parameter Behavior: Ensure consistent backend logic for how the `lang` API parameter resolves the main field content versus populating the `\*\_translations` object, ideally handled by shared database functions.

\* \* \* \* \*

11\. Scalability & Future-Proof Notes

-------------------------------------

- Standardized audit columns, clear translation strategy, and robust role synchronization improve system integrity and future capabilities.

- The architecture, with considerations for FTS and potential partitioning for `media` and `translations` tables, supports future growth.

\* \* \* \* \*

12\. Next Steps

---------------

- P1 🔴 Implement and Test Core RLS Helper Functions (`public.has\_role()`, `public.has\_role\_on\_profile()`).

- P1 🔴 Implement and Test Orphaned Translation Cleanup Triggers on `user\_roles\_master`, `media`, and other relevant tables as they are developed.

- P1 🔴 Implement and Test Role Synchronization triggers (`handle\_new\_user`, `sync\_profile\_roles\_to\_auth\_user`) and update `handle\_new\_user` to correctly initialize `last\_activity\_at`.

- P1 ⚙️ Execute Full Build Order for this module, including all DDL updates from table specs v2.1/v2.2/v2.3, deferred FK constraints, indexes, and triggers.

- P1 🟢 Seed Initial Data for `languages\_master` (v2.1) and `user\_roles\_master` (v2.1), including `icon\_identifier` and audit columns with appropriate `[ADMIN\_UUID]` placeholders.

- P2 🟠 Define and Implement consistent Population Logic for `created\_by\_profile\_id`, `updated\_by\_profile\_id` (application layer or advanced triggers).

- P2 🟠 Define and Implement update mechanism for `profiles.last\_activity\_at`.

- P2 🟠 Develop Backend Media Processing Logic (variants, checksums, etc., for `media` table).

- P2 🟠 Plan and Implement Triggers for `media.last\_linked\_or\_used\_at` on all media \*linking\* tables.

- P2 🟠 Define clear conventions for `icon\_identifier` usage and asset sources.

\* \* \* \* \*