<https://gemini.google.com/u/1/app/cc0bffb89378d0df?is_sa=1&android-min-version=301356232&ios-min-version=322.0&campaign_id=bkws&utm_source=google&utm_medium=cpc&utm_campaign=2024enUS_gemfeb&pt=9008&mt=8&ct=p-growth-sem-bkws>

https://gemini.google.com/u/1/app/e2231210a6330ec4

<https://gemini.google.com/u/1/app/a3fdff8118a11194>

### 3\. Updated Production-Ready Specification: `public.regions`

1\. Purpose & Primary Use-Cases

This table stores core identifying, geographical, and relational information for major administrative or geographical regions relevant to pilgrimage trails (e.g., Toscana, Umbria). It serves as an anchor for regional data, linking to translated descriptive content, media, and other contextual details. All textual content is managed via `public.translations`, and characteristic tags are codes from `public.characteristic\_tags\_master`.

Key User-Story Touchpoints:

- Pilgrim: Browses regions (names/details via `translations`, characteristic tags via `characteristic\_tags\_master` and `translations`).

- Admin: Manages core region records (slug, codes, links) and initiates translation. Manages `is\_featured` status.

- Manager: May contribute/verify primary language content for regional fields in `translations`.

2\. Schema

| Column Name | Data Type | Constraints | Description |

| `id` | `SERIAL` | `PRIMARY KEY` | Unique identifier for each region. Used as `row\_foreign\_key` in `translations`. |

| `slug` | `TEXT` | `NOT NULL`, `UNIQUE` | URL-friendly identifier (e.g., "tuscany"). |

| `country\_code` | `TEXT` | `NOT NULL DEFAULT 'IT'`, `REFERENCES public.countries(iso\_3166\_1\_alpha\_2) ON DELETE RESTRICT ON UPDATE CASCADE` | 2-letter ISO 3166-1 alpha-2 country code. |

| `iso\_3166\_2\_code` | `TEXT` | `UNIQUE` | ISO 3166-2 subdivision code (e.g., "IT-52"). |

| `characteristics\_tags` | `TEXT[]` | | Array of `code` values from `public.characteristic\_tags\_master`. (Translatable names via `translations`). |

| `primary\_media\_id` | `INTEGER` | `REFERENCES public.media(id) ON DELETE SET NULL` | FK to `media` table for the primary representative image. (Alt text via `translations` linked to `media`). |

| `official\_tourism\_url` | `TEXT` | `CHECK (official\_tourism\_url IS NULL OR official\_tourism\_url ~\* '^https?://.+')` | URL to the region's official tourism website. |

| `map\_default\_latitude` | `DOUBLE PRECISION` | `CHECK (map\_default\_latitude >= -90 AND map\_default\_latitude <= 90)` | Default latitude (WGS 84) for centering maps. |

| `map\_default\_longitude` | `DOUBLE PRECISION` | `CHECK (map\_default\_longitude >= -180 AND map\_default\_longitude <= 180)` | Default longitude (WGS 84) for centering maps. |

| `map\_default\_zoom` | `SMALLINT` | `CHECK (map\_default\_zoom >= 0 AND map\_default\_zoom <= 22)` | Default zoom level for maps. |

| `geo\_boundary` | `GEOMETRY(Geometry, 4326)` | | Geographical boundary (Polygon/MultiPolygon, SRID 4326). |

| `is\_featured` | `BOOLEAN` | `NOT NULL DEFAULT false` | Flag to highlight the region in UI sections. |

| `content\_visibility\_status` | `content\_visibility\_status\_enum` | `NOT NULL DEFAULT 'draft'` | Moderation status of the core region entity. |

| `created\_by\_profile\_id` | `UUID` | `REFERENCES public.profiles(id) ON DELETE SET NULL` | Profile ID of the user who created the core region record. |

| `updated\_by\_profile\_id` | `UUID` | `REFERENCES public.profiles(id) ON DELETE SET NULL` | Profile ID of the user who last updated the core region record. |

| `created\_at` | `TIMESTAMPTZ` | `NOT NULL DEFAULT now()` | Timestamp of core record creation. |

| `updated\_at` | `TIMESTAMPTZ` | `NOT NULL DEFAULT now()` | Timestamp of core record last update. Auto-updated by trigger. |

| `deleted\_at` | `TIMESTAMPTZ` | | Timestamp for soft deletion. Active records have `deleted\_at IS NULL`. |

Translatable Fields (Managed in public.translations table):

table\_identifier = 'regions', row\_foreign\_key = regions.id::text

- `name` (`column\_identifier = 'name'`)

- `description` (`column\_identifier = 'description'`)

- `cultural\_highlights` (`column\_identifier = 'cultural\_highlights'`) (Translatable via `public.translations`)

- `regional\_contact\_info` (`column\_identifier = 'regional\_contact\_info'`) (Translatable via `public.translations`)

- `best\_time\_to\_visit` (`column\_identifier = 'best\_time\_to\_visit'`) (Translatable via `public.translations`)

3\. PostgreSQL DDL

SQL

```

-- Assumed ENUM Type: content\_visibility\_status\_enum

-- Assumed dependent tables: public.countries, public.media, public.profiles, public.characteristic\_tags\_master

-- Assumed i18n tables: public.translations

CREATE TABLE IF NOT EXISTS public.regions (

id SERIAL PRIMARY KEY,

slug TEXT NOT NULL UNIQUE,

country\_code TEXT NOT NULL DEFAULT 'IT' REFERENCES public.countries(iso\_3166\_1\_alpha\_2) ON DELETE RESTRICT ON UPDATE CASCADE,

iso\_3166\_2\_code TEXT UNIQUE,

characteristics\_tags TEXT[], -- Stores codes from characteristic\_tags\_master.code

primary\_media\_id INTEGER REFERENCES public.media(id) ON DELETE SET NULL,

official\_tourism\_url TEXT,

map\_default\_latitude DOUBLE PRECISION CHECK (map\_default\_latitude >= -90 AND map\_default\_latitude <= 90),

map\_default\_longitude DOUBLE PRECISION CHECK (map\_default\_longitude >= -180 AND map\_default\_longitude <= 180),

map\_default\_zoom SMALLINT CHECK (map\_default\_zoom >= 0 AND map\_default\_zoom <= 22),

geo\_boundary GEOMETRY(Geometry, 4326),

is\_featured BOOLEAN NOT NULL DEFAULT false,

content\_visibility\_status content\_visibility\_status\_enum NOT NULL DEFAULT 'draft',

created\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL,

updated\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

deleted\_at TIMESTAMPTZ,

CONSTRAINT check\_official\_tourism\_url\_format CHECK (official\_tourism\_url IS NULL OR official\_tourism\_url ~\* '^https?://.+')

);

COMMENT ON TABLE public.regions IS 'Stores core identifying, geographical, and relational information for major regions. Textual content is in public.translations. Version: 2.0';

COMMENT ON COLUMN public.regions.id IS 'Unique identifier for each region. Used as row\_foreign\_key in translations table.';

COMMENT ON COLUMN public.regions.slug IS 'URL-friendly identifier (e.g., "tuscany"). Typically derived from the primary language name.';

COMMENT ON COLUMN public.regions.country\_code IS '2-letter ISO 3166-1 alpha-2 country code this region belongs to.';

COMMENT ON COLUMN public.regions.iso\_3166\_2\_code IS 'ISO 3166-2 subdivision code for the region (e.g., "IT-52" for Toscana).';

COMMENT ON COLUMN public.regions.characteristics\_tags IS 'Array of code values from public.characteristic\_tags\_master. Describes general features. Translated names via translations linked to characteristic\_tags\_master.';

COMMENT ON COLUMN public.regions.primary\_media\_id IS 'FK to media table for the primary representative image. Alt text for this image is in translations via media table.';

COMMENT ON COLUMN public.regions.official\_tourism\_url IS 'URL to the region''s official tourism website. Validated for basic URL format.';

COMMENT ON COLUMN public.regions.map\_default\_latitude IS 'Default latitude (WGS 84) for centering maps on this region.';

COMMENT ON COLUMN public.regions.map\_default\_longitude IS 'Default longitude (WGS 84) for centering maps on this region.';

COMMENT ON COLUMN public.regions.map\_default\_zoom IS 'Default zoom level for maps when displaying this region.';

COMMENT ON COLUMN public.regions.geo\_boundary IS 'Geographical boundary (Polygon/MultiPolygon, SRID 4326) of the region.';

COMMENT ON COLUMN public.regions.is\_featured IS 'Flag to highlight the region in UI sections (e.g., homepage).';

COMMENT ON COLUMN public.regions.content\_visibility\_status IS 'Moderation status of the core region entity (translations have their own status).';

COMMENT ON COLUMN public.regions.created\_by\_profile\_id IS 'Profile ID of the user who created the core region record.';

COMMENT ON COLUMN public.regions.updated\_by\_profile\_id IS 'Profile ID of the user who last updated the core region record.';

COMMENT ON COLUMN public.regions.created\_at IS 'Timestamp of core record creation.';

COMMENT ON COLUMN public.regions.updated\_at IS 'Timestamp of core record last update. Auto-updated by trigger.';

COMMENT ON COLUMN public.regions.deleted\_at IS 'Timestamp for soft deletion. Active records have deleted\_at IS NULL. Deleting a region will require deleting its translations via trigger.';

-- Indexing

CREATE INDEX IF NOT EXISTS idx\_regions\_slug ON public.regions(slug);

CREATE INDEX IF NOT EXISTS idx\_regions\_country\_code ON public.regions(country\_code);

CREATE INDEX IF NOT EXISTS idx\_regions\_iso\_3166\_2\_code ON public.regions(iso\_3166\_2\_code) WHERE iso\_3166\_2\_code IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_regions\_geo\_boundary ON public.regions USING GIST (geo\_boundary);

CREATE INDEX IF NOT EXISTS idx\_regions\_is\_featured ON public.regions(is\_featured);

CREATE INDEX IF NOT EXISTS idx\_regions\_content\_visibility\_status ON public.regions(content\_visibility\_status);

CREATE INDEX IF NOT EXISTS idx\_regions\_characteristics\_tags ON public.regions USING GIN (characteristics\_tags); -- For array searching

-- Trigger for updated\_at (ensure public.handle\_updated\_at() function exists)

CREATE TRIGGER trigger\_regions\_set\_updated\_at

BEFORE UPDATE ON public.regions

FOR EACH ROW

EXECUTE FUNCTION public.handle\_updated\_at();

-- Trigger for deleting related translations (ensure public.cleanup\_related\_translations() function exists)

CREATE TRIGGER trigger\_delete\_region\_translations

AFTER DELETE ON public.regions

FOR EACH ROW EXECUTE FUNCTION public.cleanup\_related\_translations(); -- Uses OLD.id::TEXT and 'regions'

-- Trigger for validating characteristics\_tags

CREATE OR REPLACE FUNCTION public.validate\_region\_characteristics\_tags()

RETURNS TRIGGER AS $$

DECLARE

tag\_code TEXT;

tag\_record public.characteristic\_tags\_master;

BEGIN

IF TG\_OP = 'INSERT' OR (TG\_OP = 'UPDATE' AND NEW.characteristics\_tags IS DISTINCT FROM OLD.characteristics\_tags) THEN

IF NEW.characteristics\_tags IS NOT NULL THEN

FOREACH tag\_code IN ARRAY NEW.characteristics\_tags

LOOP

SELECT \* INTO tag\_record FROM public.characteristic\_tags\_master WHERE code = tag\_code;

IF NOT FOUND THEN

RAISE EXCEPTION 'Invalid characteristic\_tag\_code "%" does not exist in characteristic\_tags\_master.', tag\_code;

END IF;

IF NOT tag\_record.is\_active THEN

RAISE EXCEPTION 'Characteristic tag "%" is not active.', tag\_code;

END IF;

END LOOP;

END IF;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER; -- SECURITY DEFINER if characteristic\_tags\_master has restrictive RLS for invoking user. Review implications.

CREATE TRIGGER trigger\_validate\_characteristics\_tags

BEFORE INSERT OR UPDATE ON public.regions

FOR EACH ROW

EXECUTE FUNCTION public.validate\_region\_characteristics\_tags();

-- RLS Policies

ALTER TABLE public.regions ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Allow admin full access on regions"

ON public.regions

FOR ALL

USING (public.is\_platform\_admin()) -- Assumes helper function

WITH CHECK (public.is\_platform\_admin());

CREATE POLICY "Allow authenticated users read access on regions"

ON public.regions

FOR SELECT

TO authenticated

USING (deleted\_at IS NULL AND content\_visibility\_status = 'published'); -- Example, adjust as per full requirements

CREATE POLICY "Allow anonymous users read access on regions"

ON public.regions

FOR SELECT

TO anon

USING (deleted\_at IS NULL AND content\_visibility\_status = 'published'); -- Example, adjust

```

4\. JSON Schema Mirror

JSON

```

{

"title": "region",

"description": "Core identifying and geographical information for regions. Textual content is in 'translations' table. Characteristic tags are codes from 'characteristic\_tags\_master'. Version: 2.0",

"type": "object",

"properties": {

"id": { "type": "integer", "readOnly": true, "description": "Unique identifier. Used as 'row\_foreign\_key' in 'translations'." },

"slug": { "type": "string", "description": "URL-friendly identifier (e.g., \"tuscany\"). Unique.", "pattern": "^[a-z0-9]+(?:-[a-z0-9]+)\*$", "maxLength": 255 },

"country\_code": { "type": "string", "description": "2-letter ISO 3166-1 alpha-2 country code. Defaults to 'IT'.", "default": "IT", "pattern": "^[A-Z]{2}$" },

"iso\_3166\_2\_code": { "type": ["string", "null"], "description": "ISO 3166-2 subdivision code (e.g., \"IT-52\" for Toscana). Unique if provided.", "pattern": "^[A-Z]{2}-[A-Z0-9]{1,3}$" },

"characteristics\_tags": { "type": ["array", "null"], "items": { "type": "string" }, "description": "Array of codes from public.characteristic\_tags\_master.code." },

"primary\_media\_id": { "type": ["integer", "null"], "description": "FK to 'media' table for the primary image." },

"official\_tourism\_url": { "type": ["string", "null"], "format": "url", "description": "URL to the region's official tourism website." },

"map\_default\_latitude": { "type": ["number", "null"], "minimum": -90, "maximum": 90 },

"map\_default\_longitude": { "type": ["number", "null"], "minimum": -180, "maximum": 180 },

"map\_default\_zoom": { "type": ["integer", "null"], "minimum": 0, "maximum": 22 },

"geo\_boundary": { "type": ["object", "null"], "description": "GeoJSON representation of the geometry." },

"is\_featured": { "type": "boolean", "default": false },

"content\_visibility\_status": { "type": "string", "default": "draft", "enum": ["draft", "pending\_review", "published", "archived", "rejected"], "description": "Moderation status." },

"created\_by\_profile\_id": { "type": ["string", "null"], "format": "uuid" },

"updated\_by\_profile\_id": { "type": ["string", "null"], "format": "uuid" },

"created\_at": { "type": "string", "format": "date-time", "readOnly": true },

"updated\_at": { "type": "string", "format": "date-time", "readOnly": true },

"deleted\_at": { "type": ["string", "null"], "format": "date-time", "readOnly": true }

},

"required": [ "slug", "country\_code", "is\_featured", "content\_visibility\_status" ]

}

```

5\. Relationships & Integrity

- Primary Key: `id` (SERIAL).

- Foreign Keys: `country\_code` to `public.countries`, `primary\_media\_id` to `public.media`, `created\_by\_profile\_id`/`updated\_by\_profile\_id` to `public.profiles`.

- Array FK Integrity: `characteristics\_tags` elements are validated against `public.characteristic\_tags\_master.code` (and its `is\_active` status) via `trigger\_validate\_characteristics\_tags`.

- URL Validation: `official\_tourism\_url` uses a `CHECK` constraint.

- Referenced By: `public.translations`, `public.towns.region\_id`, `public.provinces.region\_id`. Potentially `trail\_regions.region\_id`.

- Orphaned Translations: Managed by `trigger\_delete\_region\_translations`.

6\. Multilingual Strategy

All displayable text content for regions is stored in public.translations (table\_identifier = 'regions'). characteristics\_tags codes are translated by looking up their respective entries in translations (where table\_identifier = 'characteristic\_tags\_master').

7\. Role-Based Workflow & RLS Notes

- Content Management: Admins/Regional Managers manage core `regions` records. Textual content via `public.translations` workflow.

- RLS Policies: Admins full access. Authenticated/Anonymous users have read access to published and non-deleted regions. `content\_visibility\_status` and `deleted\_at` are key for RLS.

8\. ENUM vs Lookup Discussion

characteristics\_tags correctly uses codes from a master table (characteristic\_tags\_master), replacing an ENUM approach for better flexibility and i18n. content\_visibility\_status\_enum is a suitable global ENUM.

9\. UI/UX Enablement

- UI fetches core data from `public.regions` and textual content from `public.translations`.

- `characteristics\_tags` codes are used to fetch translated names and icons from `characteristic\_tags\_master` and `translations`.

- Helper functions (e.g., for translations) can simplify data retrieval for the API.

10\. Key Considerations & Definitions

- Canonical Name: Defined by entry in `public.translations` for the primary content language.

- Slug Generation: Ideally from primary language name; consider update strategy if name changes.

- Media Galleries: If multiple images per region are needed beyond `primary\_media\_id`, a `public.region\_media` linking table (with `media\_id` and `media\_role\_code`) should be implemented.

- `validate\_region\_characteristics\_tags` Security: The trigger function may need `SECURITY DEFINER` if the invoking user might not have direct SELECT permission on `characteristic\_tags\_master`, but this must be used cautiously with a hardened `search\_path`.

11\. Scalability & Future-Proofing

Centralized i18n is scalable. GIST/GIN indexes aid performance. translations table is the primary candidate for future scaling considerations like partitioning.

12\. Next-Action Checklist

- 🔴 Define Helper Function: Ensure `public.is\_platform\_admin()` is defined and secure.

- 🔴 Define `updated\_at` Trigger Function: Ensure `public.handle\_updated\_at()` is defined.

- 🔴 Define `content\_visibility\_status\_enum`: Ensure this ENUM type is created.

- 🔴 Review `validate\_region\_characteristics\_tags`: Finalize the function logic and decide on `SECURITY DEFINER` vs. `SECURITY INVOKER` based on permissions.

- 🟠 Deploy DDL: Apply the `CREATE TABLE` and `COMMENT` DDL for `public.regions`.

- 🟠 Implement Triggers: Create/apply `trigger\_regions\_set\_updated\_at`, `trigger\_delete\_region\_translations`, and `trigger\_validate\_characteristics\_tags`.

- 🟠 Implement RLS Policies: Apply the RLS policies as defined.

- 🟢 Data Migration: Plan migration if updating from an older schema (especially for `characteristics\_tags` if it was previously an enum).

- 🟢 Consider `region\_media` Table: Evaluate need for a dedicated gallery media linking table for regions.

### 4\. Diff Summary

- Added Columns:

- None (the input spec `3.1 regions.docx` was already quite complete for columns).

- Modified Columns:

- `official\_tourism\_url`: Added a `CHECK` constraint for URL format.

- Removed Columns:

- None.

- Added DDL Elements:

- `COMMENT ON TABLE` and `COMMENT ON COLUMN` statements.

- `CHECK` constraint `check\_official\_tourism\_url\_format`.

- `validate\_region\_characteristics\_tags()` function and `trigger\_validate\_characteristics\_tags` trigger.

- RLS policies updated: admin policy uses `public.is\_platform\_admin()` and `WITH CHECK`; public read policies now explicitly check `deleted\_at IS NULL` and `content\_visibility\_status = 'published'`.

- Modified DDL Elements:

- `updated\_at` trigger function reference standardized to `public.handle\_updated\_at()`.

- RLS policies updated to be more robust and aligned with checklist.

- Added Documentation/Sections:

- Section 10 ("Key Considerations & Definitions") expanded to include media galleries and trigger security.

- "Next-Action Checklist" made more specific.

- JSON Schema:

- Added `enum` to `content\_visibility\_status`.

- Included comments from DDL where appropriate.

- Notes:

- The original spec `3.1 regions.docx` was already V3 and i18n-aligned, so fewer structural changes were needed compared to a V1 spec. The focus was on V2 checklist compliance items like array FK integrity, explicit comments, and RLS hardening.