<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.towns`

1\. Purpose & Primary Use-Cases

This table stores core identifying, geographical, relational, and key non-translatable attribute data for cities, towns, villages, and significant hamlets. It's crucial for waypoints and often defines route segment start/end points. All descriptive and named textual content is managed via `public.translations`. Service tags are codes from `public.service\_tags\_master`.

Key User-Story Touchpoints:

- Pilgrim: Interacts with town information (names, descriptions, service summaries) via `translations`.

- Regional Manager: May contribute/verify primary language content in `translations`; updates core town attributes.

- Admin: Manages core town records and initiates translation.

2\. Schema

(Refer to 3.2 towns.docx, Section 2, for the detailed column list. Key attributes include id, region\_id, province\_id, slug, geospatial data, town\_type\_code, key\_services\_summary\_tags, town\_transport\_information\_urls JSONB, boolean flags, audit columns, and deleted\_at.)

Translatable Fields (Managed in public.translations table):

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

Extensive list includes name, alternate\_names (JSON array in translated\_text), short\_description, full\_description, various notes fields, major\_annual\_events\_tags (JSON array in translated\_text).

3\. PostgreSQL DDL

SQL

```

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

-- Assumed dependent tables: public.regions, public.provinces, public.town\_types\_master, public.media, public.profiles, public.service\_tags\_master

-- Assumed i18n tables: public.translations

CREATE TABLE IF NOT EXISTS public.towns (

id SERIAL PRIMARY KEY,

region\_id INTEGER REFERENCES public.regions(id) ON DELETE SET NULL,

province\_id INTEGER REFERENCES public.provinces(id) ON DELETE SET NULL,

slug TEXT NOT NULL UNIQUE,

latitude\_centroid DOUBLE PRECISION NOT NULL CHECK (latitude\_centroid >= -90 AND latitude\_centroid <= 90),

longitude\_centroid DOUBLE PRECISION NOT NULL CHECK (longitude\_centroid >= -180 AND longitude\_centroid <= 180),

geom\_centroid GEOMETRY(Point, 4326) GENERATED ALWAYS AS (ST\_SetSRID(ST\_MakePoint(longitude\_centroid, latitude\_centroid), 4326)) STORED,

geom\_boundary GEOMETRY(MultiPolygon, 4326),

elevation\_meters INTEGER,

population INTEGER CHECK (population IS NULL OR population >= 0),

town\_type\_code TEXT REFERENCES public.town\_types\_master(code) ON DELETE SET NULL,

istat\_code TEXT UNIQUE,

wikidata\_id TEXT UNIQUE,

geonames\_id INTEGER UNIQUE,

key\_services\_summary\_tags TEXT[], -- Stores codes from service\_tags\_master.code

town\_transport\_information\_urls JSONB,

has\_train\_station BOOLEAN NOT NULL DEFAULT false,

has\_bus\_services BOOLEAN NOT NULL DEFAULT false,

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

website\_url\_official\_town TEXT,

is\_major\_stage\_town BOOLEAN NOT NULL DEFAULT false,

data\_last\_verified\_at TIMESTAMPTZ,

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\_website\_url\_official\_town\_format CHECK (website\_url\_official\_town IS NULL OR website\_url\_official\_town ~\* '^https?://.+')

);

COMMENT ON TABLE public.towns IS 'Stores core identifying, geographical, and relational data for towns and settlements. All displayable text content is in public.translations. Version: 2.0';

-- Add COMMENT ON COLUMN for all columns similar to public.regions example. E.g.:

COMMENT ON COLUMN public.towns.id IS 'Primary Key. Unique identifier for each town/settlement. Used as row\_foreign\_key in translations.';

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

COMMENT ON COLUMN public.towns.geom\_centroid IS 'PostGIS Point geometry for the town''s centroid. Auto-generated from latitude\_centroid and longitude\_centroid. SRID 4326.';

COMMENT ON COLUMN public.towns.key\_services\_summary\_tags IS 'Array of code values from public.service\_tags\_master. Indicates general availability of key services. Tag display names/icons via service\_tags\_master and translations.';

COMMENT ON COLUMN public.towns.town\_transport\_information\_urls IS 'JSONB array of objects linking to transport operator websites, e.g., [{"operator\_identifier": "trenitalia\_official", "url": "..."}]. Operator name translated via translations using operator\_identifier.';

COMMENT ON COLUMN public.towns.deleted\_at IS 'Timestamp for soft deletion. Requires trigger to delete associated translations.';

-- Indexing

CREATE INDEX IF NOT EXISTS idx\_towns\_slug ON public.towns(slug); [cite: 343]

CREATE INDEX IF NOT EXISTS idx\_towns\_region\_id ON public.towns(region\_id); [cite: 344]

CREATE INDEX IF NOT EXISTS idx\_towns\_province\_id ON public.towns(province\_id); [cite: 344]

CREATE INDEX IF NOT EXISTS idx\_towns\_town\_type\_code ON public.towns(town\_type\_code); [cite: 345]

CREATE INDEX IF NOT EXISTS idx\_towns\_geom\_centroid ON public.towns USING GIST (geom\_centroid); [cite: 345]

CREATE INDEX IF NOT EXISTS idx\_towns\_geom\_boundary ON public.towns USING GIST (geom\_boundary) WHERE geom\_boundary IS NOT NULL; [cite: 346]

CREATE INDEX IF NOT EXISTS idx\_towns\_is\_major\_stage\_town ON public.towns(is\_major\_stage\_town); [cite: 347]

CREATE INDEX IF NOT EXISTS idx\_towns\_content\_visibility\_status ON public.towns(content\_visibility\_status); [cite: 348]

CREATE INDEX IF NOT EXISTS idx\_towns\_key\_services\_summary\_tags ON public.towns USING GIN (key\_services\_summary\_tags); [cite: 349]

CREATE INDEX IF NOT EXISTS idx\_towns\_town\_transport\_information\_urls\_gin ON public.towns USING GIN (town\_transport\_information\_urls); -- If querying JSONB content

-- Trigger for updated\_at

CREATE TRIGGER trigger\_towns\_set\_updated\_at

BEFORE UPDATE ON public.towns

FOR EACH ROW

EXECUTE FUNCTION public.handle\_updated\_at(); [cite: 349]

-- Trigger for deleting related translations

CREATE TRIGGER trigger\_delete\_town\_translations

AFTER DELETE ON public.towns

FOR EACH ROW EXECUTE FUNCTION public.cleanup\_related\_translations(); [cite: 350]

-- Trigger for validating key\_services\_summary\_tags

CREATE OR REPLACE FUNCTION public.validate\_town\_services\_tags()

RETURNS TRIGGER AS $$

DECLARE

tag\_code TEXT;

tag\_record public.service\_tags\_master;

BEGIN

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

IF NEW.key\_services\_summary\_tags IS NOT NULL THEN

FOREACH tag\_code IN ARRAY NEW.key\_services\_summary\_tags

LOOP

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

IF NOT FOUND THEN

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

END IF;

IF NOT tag\_record.is\_active THEN

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

END IF;

END LOOP;

END IF;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER; -- Review SECURITY DEFINER implications

CREATE TRIGGER trigger\_validate\_key\_services\_tags

BEFORE INSERT OR UPDATE ON public.towns

FOR EACH ROW

EXECUTE FUNCTION public.validate\_town\_services\_tags();

-- RLS Policies

ALTER TABLE public.towns ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Allow admin full access on towns"

ON public.towns

FOR ALL

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

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

CREATE POLICY "Allow authenticated users read access on towns"

ON public.towns

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 towns"

ON public.towns

FOR SELECT

TO anon

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

```

4\. JSON Schema Mirror

(Refer to 3.2 towns.docx, Section 4, for the detailed JSON schema. Ensure it matches the final DDL including check\_website\_url\_official\_town\_format pattern and descriptions reflect final DDL comments.)

JSON

```

{

"title": "town",

"description": "Core identifying, geographical, and relational data for towns. All displayable text content is in 'translations' table. Service tags are codes from 'service\_tags\_master'. Version: 2.0",

"type": "object",

"properties": {

"id": { "type": "integer", "readOnly": true },

"region\_id": { "type": ["integer", "null"] },

"province\_id": { "type": ["integer", "null"] },

"slug": { "type": "string", "pattern": "^[a-z0-9]+(?:-[a-z0-9]+)\*$", "maxLength": 255 },

"latitude\_centroid": { "type": "number", "minimum": -90, "maximum": 90 },

"longitude\_centroid": { "type": "number", "minimum": -180, "maximum": 180 },

"geom\_centroid": { "type": ["object", "null"], "readOnly": true, "description": "GeoJSON Point. Auto-generated." },

"geom\_boundary": { "type": ["object", "null"], "description": "GeoJSON MultiPolygon." },

"elevation\_meters": { "type": ["integer", "null"] },

"population": { "type": ["integer", "null"], "minimum": 0 },

"town\_type\_code": { "type": ["string", "null"], "description": "Code from town\_types\_master." },

"istat\_code": { "type": ["string", "null"], "pattern": "^[0-9]{6}$" },

"wikidata\_id": { "type": ["string", "null"], "pattern": "^Q[0-9]+$" },

"geonames\_id": { "type": ["integer", "null"] },

"key\_services\_summary\_tags": { "type": ["array", "null"], "items": { "type": "string" }, "description": "Array of codes from public.service\_tags\_master.code." },

"town\_transport\_information\_urls": { "type": ["array", "null"], "items": {"type": "object"} , "description": "e.g., [{\"operator\_identifier\": \"trenitalia\_official\", \"url\": \"...\"}]" },

"has\_train\_station": { "type": "boolean", "default": false },

"has\_bus\_services": { "type": "boolean", "default": false },

"primary\_media\_id": { "type": ["integer", "null"] },

"website\_url\_official\_town": { "type": ["string", "null"], "format": "url" },

"is\_major\_stage\_town": { "type": "boolean", "default": false },

"data\_last\_verified\_at": { "type": ["string", "null"], "format": "date-time" },

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

"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", "latitude\_centroid", "longitude\_centroid", "has\_train\_station", "has\_bus\_services", "is\_major\_stage\_town", "content\_visibility\_status" ]

}

```

5\. Relationships & Integrity

- Primary Key: `id` (SERIAL).

- Foreign Keys: To `regions`, `provinces`, `town\_types\_master`, `media`, `profiles`.

- Array FK Integrity: `key\_services\_summary\_tags` validated by `trigger\_validate\_key\_services\_tags`.

- URL Validation: `website\_url\_official\_town` via `CHECK` constraint.

- Generated Column: `geom\_centroid` from `latitude\_centroid`, `longitude\_centroid`.

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

6\. Multilingual Strategy

All displayable text content is in public.translations (table\_identifier = 'towns'). key\_services\_summary\_tags codes translated via service\_tags\_master and translations. JSONB town\_transport\_information\_urls uses operator\_identifier or translation\_key for internal translatable strings. Conceptual text arrays (e.g., alternate\_names) stored as JSON arrays in translations.translated\_text.

7\. Role-Based Workflow & RLS Notes

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

- RLS Policies: Admins full access. Authenticated/Anonymous users read access to published, non-deleted towns.

8\. ENUM vs Lookup Discussion

town\_type\_code references town\_types\_master. key\_services\_summary\_tags uses codes from service\_tags\_master, replacing an old ENUM.

9\. UI/UX Enablement

- UI fetches core data from `towns`, text from `translations`.

- `key\_services\_summary\_tags` used to get translated names/icons.

- Boolean flags and geo-data drive UI elements directly.

10\. Key Considerations & Definitions

- Canonical Name/Slug: Slug from primary language name.

- Data Entry: Create town record, then add translations.

- `validate\_town\_services\_tags` Security: Review `SECURITY DEFINER` need.

11\. Scalability & Future-Proofing

Centralized i18n is scalable. service\_tags\_master is flexible. Indexing aids performance.

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 is created.

- 🔴 Review `validate\_town\_services\_tags`: Finalize logic and `SECURITY DEFINER` implications.

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

- 🟠 Implement Triggers: Create/apply all specified triggers.

- 🟠 Implement RLS Policies: Apply RLS policies.

- 🟢 Data Migration: Plan migration if updating from an older schema (especially for `key\_services\_summary\_tags`).

- 🟢 Drop Old ENUM: Plan `DROP TYPE IF EXISTS public.town\_service\_tag\_enum` after migration.

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

- 🟢 JSONB Strategy: Finalize and document i18n strategy for JSONB and conceptual array translations.