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

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

### 3\. Updated Production-Ready Specification: `public.service\_tags\_master`

1\. Purpose & Primary Use-Cases

This table serves as a master lookup for key services or amenities available in towns (e.g., "pharmacy," "atm," "potable\_water"). It provides unique, stable codes and allows association with non-translatable metadata like icons, categories, or display order. All display names and descriptions for these service tags are managed in `public.translations`. This table is referenced by `public.towns.key\_services\_summary\_tags`.

Key User-Story Touchpoints:

- Pilgrim: Sees translated service tags with icons on a town's page or as filter options.

- Admin/Manager: Uses service codes to tag towns and manages their primary language names/descriptions in `translations`. Manages codes, icons, categories, and display order in this master table.

- Platform: Uses codes for filtering towns and displaying service information.

2\. Schema

| Column Name | Data Type | Constraints | Description |

| `code` | `TEXT` | `PRIMARY KEY` | Unique, stable code for the service (e.g., 'pharmacy'). Non-changing. Used as `row\_foreign\_key` in `translations`. |

| `icon\_identifier` | `TEXT` | | Optional identifier for a UI icon (e.g., 'icon-pharmacy'). |

| `category` | `TEXT` | | Optional: A category for the service tag (e.g., 'health', 'financial') for grouping. May become an FK if complexity increases. |

| `display\_order` | `INTEGER` | `NOT NULL DEFAULT 0`, `UNIQUE` | For ordering tags in UI. Ensures deterministic sorting. |

| `is\_active` | `BOOLEAN` | `NOT NULL DEFAULT true` | Whether the tag is active and usable. |

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

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

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

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

Translatable Fields (Managed in public.translations table):

The following conceptual fields for a service tag will have their content stored in public.translations with table\_identifier = 'service\_tags\_master' and row\_foreign\_key = service\_tags\_master.code:

- `name` (e.g., `column\_identifier = 'name'`)

- `description` (e.g., `column\_identifier = 'description'`, optional)

3\. PostgreSQL DDL

SQL

```

CREATE TABLE IF NOT EXISTS public.service\_tags\_master (

code TEXT PRIMARY KEY,

icon\_identifier TEXT,

category TEXT, -- Consider FK to a service\_tag\_categories\_master if categories become complex or need their own attributes/translations.

display\_order INTEGER NOT NULL DEFAULT 0 UNIQUE,

is\_active BOOLEAN NOT NULL DEFAULT true,

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

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

);

COMMENT ON TABLE public.service\_tags\_master IS 'Master lookup for town service tags (amenities). Names/descriptions are in public.translations. Referenced by towns.key\_services\_summary\_tags. Version: 2.0';

COMMENT ON COLUMN public.service\_tags\_master.code IS 'Primary Key. Unique, stable code for the service tag (e.g., ''pharmacy'', ''atm''). Non-changing. Used as row\_foreign\_key in translations.';

COMMENT ON COLUMN public.service\_tags\_master.icon\_identifier IS 'Optional identifier for a UI icon associated with this service (e.g., ''icon-pharmacy'', ''icon-atm'').';

COMMENT ON COLUMN public.service\_tags\_master.category IS 'Optional: A category for the service tag (e.g., ''health'', ''financial'', ''food\_drink'') for grouping in UI or filtering.';

COMMENT ON COLUMN public.service\_tags\_master.display\_order IS 'For ordering tags in UI dropdowns, filter lists, or displays. Ensures deterministic sorting.';

COMMENT ON COLUMN public.service\_tags\_master.is\_active IS 'Indicates if the service tag is active and available for use.';

COMMENT ON COLUMN public.service\_tags\_master.created\_at IS 'Timestamp of when the service tag record was created.';

COMMENT ON COLUMN public.service\_tags\_master.updated\_at IS 'Timestamp of when the service tag record was last updated. Auto-updated by a trigger.';

COMMENT ON COLUMN public.service\_tags\_master.created\_by\_profile\_id IS 'Profile ID of the user who created the service tag record.';

COMMENT ON COLUMN public.service\_tags\_master.updated\_by\_profile\_id IS 'Profile ID of the user who last updated the service tag record.';

-- Indexing:

-- PRIMARY KEY on 'code' already creates a unique index. [cite: 569]

-- UNIQUE constraint on 'display\_order' already creates an index. [cite: 569]

CREATE INDEX IF NOT EXISTS idx\_service\_tags\_master\_category ON public.service\_tags\_master(category) WHERE category IS NOT NULL; -- For filtering/grouping by category [cite: 570]

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

CREATE TRIGGER trigger\_service\_tags\_master\_set\_updated\_at

BEFORE UPDATE ON public.service\_tags\_master

FOR EACH ROW

EXECUTE FUNCTION public.handle\_updated\_at(); -- Or extensions.moddatetime, or public.set\_current\_timestamp\_updated\_at

-- Trigger for deleting related translations

-- (ensure public.cleanup\_related\_translations() function is adaptable for TEXT PKs

-- or a specific version is made that correctly uses OLD.code as the row\_foreign\_key

-- for table\_identifier = 'service\_tags\_master')

CREATE TRIGGER trigger\_delete\_service\_tag\_translations

AFTER DELETE ON public.service\_tags\_master

FOR EACH ROW EXECUTE FUNCTION public.cleanup\_related\_translations();

-- RLS Policies

ALTER TABLE public.service\_tags\_master ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Allow admin full access on service\_tags\_master"

ON public.service\_tags\_master

FOR ALL

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

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

CREATE POLICY "Allow authenticated users read access on service\_tags\_master"

ON public.service\_tags\_master

FOR SELECT

TO authenticated

USING (is\_active = true);

CREATE POLICY "Allow anonymous users read access on service\_tags\_master"

ON public.service\_tags\_master

FOR SELECT

TO anon

USING (is\_active = true);

```

4\. JSON Schema Mirror

JSON

```

{

"title": "service\_tag\_master",

"description": "Master lookup for town service tags. Names/descriptions are in 'translations' table. Version: 2.0",

"type": "object",

"properties": {

"code": {

"type": "string",

"description": "Primary Key. Unique, stable code for the service tag (e.g., 'pharmacy'). Used as 'row\_foreign\_key' in 'translations'.",

"pattern": "^[a-z0-9\_]+$"

},

"icon\_identifier": {

"type": ["string", "null"],

"description": "Optional identifier for a UI icon.",

"maxLength": 50

},

"category": {

"type": ["string", "null"],

"description": "Optional category for grouping service tags.",

"maxLength": 50

},

"display\_order": {

"type": "integer",

"default": 0,

"description": "Order for displaying tags in UI. Must be unique."

},

"is\_active": {

"type": "boolean",

"default": true,

"description": "Indicates if the service tag is active and available for use."

},

"created\_at": {

"type": "string",

"format": "date-time",

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"readOnly": true

},

"created\_by\_profile\_id": {

"type": ["string", "null"],

"format": "uuid",

"description": "Profile ID of the user who created the record."

},

"updated\_by\_profile\_id": {

"type": ["string", "null"],

"format": "uuid",

"description": "Profile ID of the user who last updated the record."

}

},

"required": [

"code",

"display\_order",

"is\_active"

]

}

```

5\. Relationships & Integrity

- Primary Key: `code` (TEXT).

- Referenced By:

- `public.towns.key\_services\_summary\_tags` (`TEXT[]` of codes from this table).

- `public.translations.row\_foreign\_key` (conceptually, when `table\_identifier = 'service\_tags\_master'`). Managed by trigger and application logic.

- Unique Constraints: `code` (PK), `display\_order`.

- Index: `idx\_service\_tags\_master\_category` on `category` for efficient filtering/grouping.

6\. Impact on towns Table

The towns table's key\_services\_summary\_tags column must be TEXT[]. Data migration from town\_service\_tag\_enum[] will be needed. The town\_service\_tag\_enum type will be dropped post-migration.

7\. Multilingual Strategy

Display names/descriptions are in public.translations. To fetch a translated name:

SQL

```

SELECT translated\_text

FROM public.translations

WHERE table\_identifier = 'service\_tags\_master'

AND column\_identifier = 'name' -- Or 'description'

AND row\_foreign\_key = 'tag\_code\_value' -- e.g., 'pharmacy'

AND language\_code = 'user\_language\_code'

AND translation\_status = 'published\_live'; -- Or relevant status

```

8\. Role-Based Workflow & RLS Notes

- Content Management: Core records (`code`, `icon\_identifier`, etc.) managed by Admins. Textual names/descriptions via `public.translations` workflow.

- RLS Policies: Admins full access (via `public.is\_platform\_admin()`). Authenticated/Anonymous read access to `is\_active = true` tags. `public.translations` RLS governs text access.

9\. ENUM vs Lookup Discussion

This table correctly implements the lookup pattern, replacing town\_service\_tag\_enum for richer attributes and i18n.

10\. UI/UX Enablement

- UI uses `code` for logic/translation fetching. `icon\_identifier`, `category`, `display\_order` for presentation.

- Translated `name`/`description` from `public.translations` are displayed.

- Enables filtering towns by services.

11\. Scalability & Future-Proofing

Scalable for languages and flexible for future non-translatable attributes. Table size expected to be small. The category field provides simple grouping; if categories require their own metadata or translation, they could be promoted to a separate master table.

12\. Next-Action Checklist

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

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

- 🔴 Verify/Adapt `cleanup\_related\_translations`: Confirm function handles `TEXT` PKs (`OLD.code`) for `service\_tags\_master`.

- 🟠 Deploy DDL: Apply `CREATE TABLE` and `COMMENT` DDL for `public.service\_tags\_master`.

- 🟠 Populate Initial Data: Insert seed data (e.g., `{'pharmacy', 'icon-rx', 'health', 10, true}`).

- 🟠 Populate Translations: For each seed `code`, create `name`/`description` entries in `public.translations` for supported languages.

- 🟠 Implement Triggers: Create `trigger\_service\_tags\_master\_set\_updated\_at` and `trigger\_delete\_service\_tag\_translations`.

- 🟠 Implement RLS Policies: Apply defined RLS policies.

- 🟢 Update `towns` Table: Plan `towns.key\_services\_summary\_tags` modification to `TEXT[]` and data migration.

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

### 4\. Diff Summary

- Added Columns:

- `is\_active BOOLEAN NOT NULL DEFAULT true`

- `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`

- Modified Columns:

- `category TEXT` - Added comment regarding potential future FK.

- Removed Columns:

- None.

- Added DDL Elements:

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

- RLS `WITH CHECK` clause for admin write policy.

- RLS policies for authenticated/anonymous now filter by `is\_active = true`.

- Modified DDL Elements:

- Admin RLS policy uses `public.is\_platform\_admin()`.

- `updated\_at` trigger function reference standardized.

- Added Documentation/Sections:

- Enhanced details in all 12 sections of the specification.

- "Next-Action Checklist" now includes severity icons.

- JSON Schema:

- Added `is\_active`, `created\_by\_profile\_id`, `updated\_by\_profile\_id`.

- Updated main description to include version.

- `is\_active` added to `required` properties.