<https://gemini.google.com/u/1/app/bf28f389cd093e55?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/1d4834db0819faf2>

<https://gemini.google.com/u/1/app/d423d51e9f40da78>

### 3\. Updated Production-Ready Specification (`shop\_service\_types\_master`)

This document details the structure, purpose, and considerations for the `shop\_service\_types\_master` table, Version 1.2. This version incorporates full V2 audit columns, an `is\_active` flag for lifecycle management, and refined indexing and translation handling.

1\. Purpose & Primary Use-Cases

The `shop\_service\_types\_master` table defines the various types of shops and practical services available to pilgrims (e.g., "Pharmacy," "ATM," "Grocery Store," "Post Office," "Tourist Information"). Its purpose is to provide a standardized, translatable classification that drives UI elements (map icons, filter options), helps users quickly identify available services, and ensures data consistency. Key uses include pilgrim service discovery through filtering and map icons, and admin content categorization for service-providing waypoints.

2\. Schema

| column | data\_type | constraints | description |

| `id` | `INTEGER` | Primary Key (Generated as identity always) | Unique identifier for the shop/service type. |

| `code` | `TEXT` | Unique, Not Null, CHECK (length(`code`) > 0 AND length(`code`) &lt;= 50 AND `code` ~ '^[a-z0-9\_]+$') | Short, stable, machine-readable code (e.g., 'pharmacy\_farmacia', 'atm\_bancomat'). Snake\_case. |

| `label` | `TEXT` | Not Null, CHECK (length(`label`) > 0 AND length(`label`) &lt;= 100) | Human-readable label for UI display. Primary reference language (English) text. (Translatable via `public.translations`). |

| `description` | `TEXT` | Nullable | Optional description of the shop/service type, providing more context. Primary reference language (English) text. (Translatable via `public.translations`). |

| `icon\_identifier` | `TEXT` | Nullable, CHECK (`icon\_identifier` IS NULL OR length(`icon\_identifier`) &lt;= 100) | Name, class, or path for a UI icon associated with this service type. |

| `category` | `TEXT` | Nullable, CHECK (`category` IS NULL OR length(`category`) &lt;= 50 AND `category` ~ '^[a-zA-Z0-9\_ -]\*$') | Optional broad category for grouping services (e.g., "Health", "Financial", "Groceries", "Information"). Allows letters, numbers, underscore, hyphen, space. Max 50 chars. |

| `sort\_order` | `INTEGER` | Not Null, Default `0` | Determines the display order in UI lists or filters. Lower numbers appear first. |

| `is\_active` | `BOOLEAN` | Not Null, Default `true` | True if the type is active and available for use; false if retired. |

| `created\_at` | `TIMESTAMPTZ` | Not Null, Default `now()` | Timestamp of record creation. |

| `updated\_at` | `TIMESTAMPTZ` | Not Null, Default `now()` | Timestamp of last update (auto-updated by trigger). |

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

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

3\. PostgreSQL DDL

SQL

```

-- Ensure prerequisite tables are created first:

-- public.profiles (UUID id PK)

-- public.translations (for i18n)

CREATE TABLE public.shop\_service\_types\_master (

id INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

code TEXT UNIQUE NOT NULL CHECK (length(code) > 0 AND length(code) <= 50 AND code ~ '^[a-z0-9\_]+$'),

label TEXT NOT NULL CHECK (length(label) > 0 AND length(label) <= 100),

description TEXT NULL,

icon\_identifier TEXT NULL CHECK (icon\_identifier IS NULL OR length(icon\_identifier) <= 100),

category TEXT NULL CHECK (category IS NULL OR (length(category) > 0 AND length(category) <= 50 AND category ~ '^[a-zA-Z0-9\_ -]\*$')),

sort\_order INTEGER NOT NULL DEFAULT 0,

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 NULL REFERENCES public.profiles(id) ON DELETE SET NULL,

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

);

COMMENT ON TABLE public.shop\_service\_types\_master IS 'Master list of shop and practical service types (e.g., Pharmacy, ATM). `label` and `description` are translatable. Version 1.2';

COMMENT ON COLUMN public.shop\_service\_types\_master.id IS 'Unique identifier for the shop/service type.';

COMMENT ON COLUMN public.shop\_service\_types\_master.code IS 'Short, stable, machine-readable code (snake\_case). Max 50 chars. E.g., ''pharmacy\_farmacia'', ''atm\_bancomat''.';

COMMENT ON COLUMN public.shop\_service\_types\_master.label IS 'Human-readable label for UI display. Primary reference language (English) text. (Translatable via public.translations). Max 100 chars.';

COMMENT ON COLUMN public.shop\_service\_types\_master.description IS 'Optional description of the service type. Primary reference language (English) text. (Translatable via public.translations).';

COMMENT ON COLUMN public.shop\_service\_types\_master.icon\_identifier IS 'Name, class, or path for a UI icon associated with this service type. Max 100 chars.';

COMMENT ON COLUMN public.shop\_service\_types\_master.category IS 'Optional broad category for grouping services (e.g., "Health", "Financial", "Groceries", "Information"). Allows letters, numbers, underscore, hyphen, space. Max 50 chars.';

COMMENT ON COLUMN public.shop\_service\_types\_master.sort\_order IS 'Determines the display order in UI lists or filters. Lower numbers appear first.';

COMMENT ON COLUMN public.shop\_service\_types\_master.is\_active IS 'True if the type is active and available for use; false if retired. Defaults to true.';

COMMENT ON COLUMN public.shop\_service\_types\_master.created\_at IS 'Timestamp of record creation.';

COMMENT ON COLUMN public.shop\_service\_types\_master.updated\_at IS 'Timestamp of last update (auto-updated by trigger).';

COMMENT ON COLUMN public.shop\_service\_types\_master.created\_by\_profile\_id IS 'Profile ID of the user who created the record. FK to profiles.id.';

COMMENT ON COLUMN public.shop\_service\_types\_master.updated\_by\_profile\_id IS 'Profile ID of the user who last updated the record. FK to profiles.id.';

-- Triggers & Functions

CREATE TRIGGER trigger\_shop\_service\_types\_master\_set\_updated\_at

BEFORE UPDATE ON public.shop\_service\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at(); -- Assuming this standard function exists

COMMENT ON TRIGGER trigger\_shop\_service\_types\_master\_set\_updated\_at ON public.shop\_service\_types\_master IS 'Trigger to automatically update updated\_at timestamp on row modification.';

CREATE OR REPLACE FUNCTION public.cleanup\_shop\_service\_type\_translations()

RETURNS TRIGGER AS $$

BEGIN

DELETE FROM public.translations

WHERE table\_identifier = 'shop\_service\_types\_master'

AND row\_foreign\_key = OLD.id::TEXT;

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_shop\_service\_type\_translations

AFTER DELETE ON public.shop\_service\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_shop\_service\_type\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_shop\_service\_type\_translations ON public.shop\_service\_types\_master IS 'Cleans up orphaned translations from public.translations when a shop\_service\_types\_master record is deleted.';

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_sstm\_is\_active ON public.shop\_service\_types\_master(is\_active);

CREATE INDEX IF NOT EXISTS idx\_sstm\_sort\_order ON public.shop\_service\_types\_master(sort\_order);

CREATE INDEX IF NOT EXISTS idx\_sstm\_category ON public.shop\_service\_types\_master(category) WHERE category IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_sstm\_created\_by ON public.shop\_service\_types\_master(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_sstm\_updated\_by ON public.shop\_service\_types\_master(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

-- RLS Policies

ALTER TABLE public.shop\_service\_types\_master ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Allow public read access to active shop/service types"

ON public.shop\_service\_types\_master FOR SELECT

USING (is\_active = true);

CREATE POLICY "Allow admins and managers to manage shop/service types"

ON public.shop\_service\_types\_master FOR ALL

USING (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'admin\_platform')) OR

(SELECT public.has\_role\_on\_profile(auth.uid(), 'manager')) -- Assuming 'manager' is a defined role with rights to edit this

) WITH CHECK (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'admin\_platform')) OR

(SELECT public.has\_role\_on\_profile(auth.uid(), 'manager'))

);

```

4\. JSON Schema Mirror

(Reflects the schema table in Section 2)

JSON

```

{

"title": "shop\_service\_type\_master",

"description": "Master list of shop and practical service types (e.g., Pharmacy, ATM). `label` and `description` are translatable. Version 1.2",

"type": "object",

"properties": {

"id": {

"type": "integer",

"description": "Unique identifier for the shop/service type. Primary Key.",

"readOnly": true

},

"code": {

"type": "string",

"description": "Short, stable, machine-readable code (snake\_case). Max 50 chars. E.g., 'pharmacy\_farmacia'.",

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

"maxLength": 50

},

"label": {

"type": "string",

"description": "Human-readable label for UI display. Primary reference language (English) text. (Translatable via public.translations). Max 100 chars.",

"maxLength": 100

},

"description": {

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

"description": "Optional description of the service type. Primary reference language (English) text. (Translatable via public.translations)."

},

"icon\_identifier": {

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

"maxLength": 100,

"description": "Name, class, or path for a UI icon associated with this service type."

},

"category": {

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

"maxLength": 50,

"pattern": "^[a-zA-Z0-9\_ -]\*$",

"description": "Optional broad category for grouping services (e.g., \"Health\", \"Financial\", \"Groceries\", \"Information\"). Allows letters, numbers, underscore, hyphen, space."

},

"sort\_order": {

"type": "integer",

"default": 0,

"description": "Determines the display order in UI lists or filters. Lower numbers appear first."

},

"is\_active": {

"type": "boolean",

"default": true,

"description": "True if the type is active and available for use; false if retired."

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of record creation.",

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of last update (auto-updated by trigger).",

"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",

"label",

"sort\_order",

"is\_active",

"created\_at",

"updated\_at"

// Audit user FKs are nullable

]

}

```

5\. Relationships & Integrity

- Primary Key: `id` (`INTEGER`)

- Unique Constraint: `code` must be unique and follow the defined pattern.

- Foreign Key References FROM other tables:

- `shops\_and\_services\_details.service\_type\_id` will reference `shop\_service\_types\_master.id` (ON DELETE RESTRICT). The detail table's trigger will ensure this referenced record is active.

- Foreign Key References TO other tables:

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

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

- Data Integrity Notes: The `category` field has a `CHECK` constraint for its format if used.

6\. Multilingual Strategy

- Translatable Fields:

- `label`: The primary reference language (English) label for the service type. (Translatable via `public.translations`)

- `description`: The optional primary reference language (English) description for the service type. (Translatable via `public.translations`)

- Non-Translatable Fields:

- `code` is a stable system identifier and is not translated.

- `category` text might be a code itself, or if its display needs translation, a separate mechanism or convention would be required (e.g., translating the category string directly if it's from a controlled list). For V1.2, `category` itself is not marked for direct `public.translations` linking.

- Orphan Cleanup: The `trigger\_cleanup\_shop\_service\_type\_translations` (`AFTER DELETE`) calls `public.cleanup\_related\_translations` to remove orphaned entries from `public.translations` when a `shop\_service\_types\_master` record is deleted.

7\. Role-Based Workflow & RLS Notes

- Content Management: This table is typically managed by Platform Administrators or senior Content Managers who define the canonical list of service types.

- Lifecycle: Service types are "retired" by setting `is\_active = false`. Physical deletion of a record is restricted by the Foreign Key constraint from `shops\_and\_services\_details` if any service detail entry is actively using that type.

- RLS Policies:

- Public Users (`SELECT`): Can read all `is\_active = true` service types.

- Authenticated Users (`admin\_platform`, `manager`): Have `ALL` permissions (CRUD) to manage service types, subject to their roles defined via `public.has\_role\_on\_profile`.

8\. ENUM vs Lookup Discussion

- Decision: This `shop\_service\_types\_master` table correctly promotes an original `shop\_service\_type\_enum` concept.

- Reasoning: Using a master lookup table is superior to a hardcoded ENUM for this purpose. It provides:

- Richness: Allows additional attributes like `description`, `icon\_identifier`, `category`, and `sort\_order`.

- Internationalization (i18n): `label` and `description` are designed to be translatable.

- Maintainability: Easier to add, retire (`is\_active`), or modify service types without complex database migrations.

- Data Integrity: The `code` field provides a stable, constrained identifier.

- Auditability: Standard audit columns (`created\_at`, `updated\_at`, `created\_by\_profile\_id`, `updated\_by\_profile\_id`) track changes.

9\. UI/UX Enablement

- `label` (translated): Used for display names in filters, legends, and service detail views.

- `icon\_identifier`: Drives the display of map icons and list icons, aiding quick user identification of service types.

- `category`: Can be used by the UI to group related services in filter UIs or informational directories (e.g., "Health Services," "Financial Services").

- `sort\_order`: Ensures a consistent and logical presentation order of service types in UI lists and filter options.

- `is\_active`: UI should primarily display and allow selection of active service types when new service details are being created or when users are filtering.

10\. Key Considerations & Definitions

- Comprehensiveness vs. Specificity: The initial list of service types (provided in seed data) should be comprehensive enough to cover common pilgrim needs but not so overly granular that it becomes difficult for content managers to choose or for users to navigate. The "other\_..." types provide a necessary fallback.

- Icon Strategy: A clear and consistent set of UI assets or classes must correspond to the `icon\_identifier` values to ensure visual cues are effective.

- `category` Field Usage: If the `category` field is to be used extensively for UI grouping, the values for `category` should be standardized (e.g., defined within documentation or a simple ENUM type if the list of categories becomes fixed and needs DB enforcement). For V1.2, it's a flexible text field.

- Translations for `category`: If the `category` strings themselves need to be translated for display in the UI (e.g. "Health" -> "Salute"), these strings would need to be added to the `public.translations` table with a suitable `table\_identifier` (e.g., 'shop\_service\_types\_master\_category\_values') and `row\_foreign\_key` (the category string itself, like 'Health').

11\. Scalability & Future-Proofing

- Lookup Table Structure: This structure is highly scalable. New service types can be added easily, and existing ones can be modified (e.g., changing icons or descriptions) or retired (`is\_active = false`) without requiring schema changes to the dependent `shops\_and\_services\_details` table.

- Manageable List: The number of distinct shop and service types is expected to be relatively small to moderate, ensuring good query performance.

- Audit Fields & Lifecycle: Full audit columns and the `is\_active` flag provide robust change history and allow for proper lifecycle management of service types.

12\. Next-Action Checklist

- 🔴 Create Table: Execute the DDL for `public.shop\_service\_types\_master` (Version 1.2).

- 🔴 Apply Triggers: Ensure the `set\_current\_timestamp\_updated\_at` (standard function) and `cleanup\_shop\_service\_type\_translations` triggers are created and applied.

- 🔴 Initial Population / Seed Data: Insert the defined shop/service types (as provided in the DDL section or an expanded list relevant to the project). Ensure `is\_active = true` and `created\_by\_profile\_id` (set to a system administrator's profile ID) are correctly populated for this initial seed data.

- 🟠 RLS Helper Functions: Verify that any RLS helper functions referenced (e.g., `public.has\_role\_on\_profile`) are defined, secure, and function as expected.

- 🟠 RLS Policies: Implement and thoroughly test the RLS policies.

- 🟢 Categorization Values: Review the values used in the `category` field in the seed data. If this field will be important for UI filtering, consider standardizing these category values (e.g., in project documentation or by promoting `category` to its own small lookup table if it needs descriptions or translations itself).

- 🟢 Iconography Coordination: Ensure the list of `icon\_identifier` values is coordinated with the UI/UX team for asset preparation and consistent visual representation.

- 🟢 Translation Entries: Prepare initial English entries for all translatable fields (`label`, `description`) in the `public.translations` table, correctly linked to `shop\_service\_types\_master` records.