<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/1d4834db0819faf2)

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

### 3\. Updated Production-Ready Specification (`food\_water\_source\_types\_master`)

This document details the structure, purpose, and considerations for the `food\_water\_source\_types\_master` table, Version 1.2.1 (reflecting RLS role name alignment). 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 `food\_water\_source\_types\_master` table defines the specific types of food establishments or water sources available (e.g., "Public Fountain - Potable," "Restaurant," "Grocery Store"). Its purpose is to provide a standardized, translatable classification that drives UI elements (like map icons, filter options), helps users distinguish between commercial services and free resources, and ensures data consistency. Key uses include pilgrim POI identification and filtering, admin content categorization for food/water waypoints, and system UI display of icons and filter controls.

2\. Schema

| column | data\_type | constraints | description |

| `id` | `INTEGER` | Primary Key (Generated as identity always) | Unique identifier for the food/water source 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., 'public\_fountain\_potable', 'restaurant'). 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 source 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 source type. |

| `is\_commercial` | `BOOLEAN` | Not Null, Default `false` | Flag to distinguish commercial establishments (e.g., restaurant, shop) from non-commercial ones (e.g., public fountain, natural spring). |

| `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, roles TEXT[])

-- public.translations (for i18n)

CREATE TABLE public.food\_water\_source\_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),

is\_commercial BOOLEAN NOT NULL DEFAULT FALSE,

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.food\_water\_source\_types\_master IS 'Master list of food and water source types (e.g., public fountain, restaurant). `label` and `description` are translatable. Version 1.2.1';

COMMENT ON COLUMN public.food\_water\_source\_types\_master.id IS 'Unique identifier for the food/water source type.';

COMMENT ON COLUMN public.food\_water\_source\_types\_master.code IS 'Short, stable, machine-readable code (snake\_case). Max 50 chars. E.g., ''public\_fountain\_potable''.';

COMMENT ON COLUMN public.food\_water\_source\_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.food\_water\_source\_types\_master.description IS 'Optional description of the source type. Primary reference language (English) text. (Translatable via public.translations).';

COMMENT ON COLUMN public.food\_water\_source\_types\_master.icon\_identifier IS 'Name, class, or path for a UI icon. Max 100 chars.';

COMMENT ON COLUMN public.food\_water\_source\_types\_master.is\_commercial IS 'True for commercial establishments (restaurant, shop), False for public/natural sources.';

COMMENT ON COLUMN public.food\_water\_source\_types\_master.sort\_order IS 'Determines the display order in UI lists or filters.';

COMMENT ON COLUMN public.food\_water\_source\_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.food\_water\_source\_types\_master.created\_at IS 'Timestamp of record creation.';

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

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

COMMENT ON COLUMN public.food\_water\_source\_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\_food\_water\_source\_types\_master\_set\_updated\_at

BEFORE UPDATE ON public.food\_water\_source\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at();

COMMENT ON TRIGGER trigger\_food\_water\_source\_types\_master\_set\_updated\_at ON public.food\_water\_source\_types\_master IS 'Trigger to automatically update updated\_at timestamp on row modification.';

CREATE OR REPLACE FUNCTION public.cleanup\_food\_water\_source\_type\_translations()

RETURNS TRIGGER AS $$

BEGIN

DELETE FROM public.translations

WHERE table\_identifier = 'food\_water\_source\_types\_master'

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

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_food\_water\_source\_type\_translations

AFTER DELETE ON public.food\_water\_source\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_food\_water\_source\_type\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_food\_water\_source\_type\_translations ON public.food\_water\_source\_types\_master IS 'Cleans up orphaned translations from public.translations when a food\_water\_source\_types\_master record is deleted.';

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_fwstm\_is\_active ON public.food\_water\_source\_types\_master(is\_active);

CREATE INDEX IF NOT EXISTS idx\_fwstm\_sort\_order ON public.food\_water\_source\_types\_master(sort\_order);

CREATE INDEX IF NOT EXISTS idx\_fwstm\_is\_commercial ON public.food\_water\_source\_types\_master(is\_commercial);

CREATE INDEX IF NOT EXISTS idx\_fwstm\_created\_by ON public.food\_water\_source\_types\_master(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_fwstm\_updated\_by ON public.food\_water\_source\_types\_master(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

-- RLS Policies

ALTER TABLE public.food\_water\_source\_types\_master ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Allow public read access to active food/water source types"

ON public.food\_water\_source\_types\_master FOR SELECT

USING (is\_active = true);

CREATE POLICY "Allow platform\_admins to manage food/water source types"

ON public.food\_water\_source\_types\_master FOR ALL

USING (

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

) WITH CHECK (

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

);

```

4\. JSON Schema Mirror

(Reflects the schema table in Section 2)

JSON

```

{

"title": "food\_water\_source\_type\_master",

"description": "Master list of food and water source types (e.g., public fountain, restaurant). `label` and `description` are translatable. Version 1.2.1",

"type": "object",

"properties": {

"id": {

"type": "integer",

"description": "Unique identifier for the food/water source type. Primary Key.",

"readOnly": true

},

"code": {

"type": "string",

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

"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 source 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 source type."

},

"is\_commercial": {

"type": "boolean",

"default": false,

"description": "Flag to distinguish commercial establishments from non-commercial ones."

},

"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",

"is\_commercial",

"sort\_order",

"is\_active",

"created\_at",

"updated\_at"

]

}

```

5\. Relationships & Integrity

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

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

- Foreign Key References FROM other tables: `food\_water\_sources\_details.source\_type\_id` references `food\_water\_source\_types\_master.id` (ON DELETE RESTRICT). The `food\_water\_sources\_details` table includes a trigger to ensure the referenced `food\_water\_source\_types\_master` record has `is\_active = true`.

- 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.

6\. Multilingual Strategy

- Translatable Fields: `label` and `description`. These columns store their content in the primary reference language (English). They are designated as translatable via the central `public.translations` table, where translations into other languages are stored.

- Non-Translatable Fields: `code` is a stable system identifier. `is\_commercial`, `sort\_order`, and `is\_active` are structural/flag fields and are not translatable text. `icon\_identifier` is a system value.

- Orphan Cleanup: An `AFTER DELETE` trigger (`trigger\_cleanup\_food\_water\_source\_type\_translations`) is implemented. It calls `public.cleanup\_food\_water\_source\_type\_translations` (which internally would use a generic like `public.cleanup\_related\_translations`) to remove any orphaned entries from `public.translations` when a `food\_water\_source\_types\_master` record is deleted.

7\. Role-Based Workflow & RLS Notes

- Content Management: This master table is typically managed by `platform\_admin` users who define and maintain the canonical list of food and water source types for the entire platform.

- Lifecycle: Types are made inactive or "retired" by setting their `is\_active` flag to `false`. Physical deletion of a record is restricted by the Foreign Key constraint from the `food\_water\_sources\_details` table if any food/water source detail entry is actively using that type. The "active check" trigger on `food\_water\_sources\_details` also prevents linking to inactive types.

- RLS Policies:

- Row-Level Security is enabled.

- Public Users (`SELECT`): Can read all `food\_water\_source\_types\_master` records where `is\_active = true`.

- Authenticated Users (`platform\_admin` role via `public.has\_role\_on\_profile` helper): Have `ALL` permissions (INSERT, SELECT, UPDATE, DELETE) to manage these types.

8\. ENUM vs Lookup Discussion

- Decision: This `food\_water\_source\_types\_master` table is a V2-compliant lookup table, correctly promoting an original `food\_water\_source\_type\_enum` concept.

- Reasoning: Using a lookup table is essential for this classification due to its direct impact on user safety (e.g., distinguishing potable water sources) and logistical planning (e.g., identifying commercial vs. non-commercial sources). This approach allows for richer attributes such as translatable `label` and `description` fields, a dedicated `icon\_identifier` for UI consistency, the critical `is\_commercial` boolean flag (which can drive UI logic in dependent tables), an `is\_active` flag for proper lifecycle management, `sort\_order` for controlled presentation, and standard audit columns for traceability. This offers significantly more flexibility, maintainability, and data integrity than a hardcoded ENUM.

9\. UI/UX Enablement

- `label` (translated): Used for clear and understandable display names in filters, map legends, and within the detail views of individual food/water sources.

- `icon\_identifier`: Drives the display of distinct map icons and list icons, helping users quickly identify the nature of a food or water source (e.g., a fountain icon, a restaurant icon, a grocery store icon).

- `is\_commercial`: This flag is crucial for the UI. It can be used to visually differentiate commercial points from public/free resources (e.g., different color map pins, or by determining whether to display price range information from the `food\_water\_sources\_details` table).

- `sort\_order`: Ensures a logical and consistent presentation order of these source types in UI filter dropdowns, selection lists, or legends.

- `is\_active`: The UI should primarily present and allow filtering or selection based on active source types for general users and when content managers are creating new food/water source entries.

10\. Key Considerations & Definitions

- Clarity and Distinction: The `code` and `label` for each type must be intuitive and clearly distinct to avoid any ambiguity for data managers entering information and for pilgrims interpreting it on the frontend. The distinction between various types of potable and non-potable water sources, as well as different kinds of food establishments, is particularly important.

- Icon Set: A comprehensive and easily understandable set of icons corresponding to each `icon\_identifier` needs to be available and consistently applied in the frontend application to ensure effective visual communication.

- `is\_commercial` Flag Impact: This boolean flag is a key differentiator. UI logic for displaying details from the `food\_water\_sources\_details` table should adapt based on this flag; for example, fields related to pricing or payment methods are generally only relevant if `is\_commercial = true`.

- Seed Data Comprehensiveness: The initial seed data provided for this table must be reviewed to ensure it covers all expected types of food and water sources relevant to the pilgrimage routes and user needs from the outset.

11\. Scalability & Future-Proofing

- Lookup Table Structure: The design is highly scalable. New source types can be easily added, or attributes of existing ones (like icons or descriptions) modified, without requiring schema changes to the `food\_water\_sources\_details` table or other parts of the system that might reference these types.

- Manageable Number of Types: The total number of distinct food and water source types is expected to be manageable, ensuring that the table remains performant for queries and easy to administer.

- Audit and Lifecycle Management: The inclusion of full audit columns and the `is\_active` flag provides a robust framework for tracking changes and managing the lifecycle of these source types, allowing types to be retired gracefully without losing historical data context.

12\. Next-Action Checklist

- 🔴 Verify Prerequisite Tables: Confirm `public.profiles` and `public.translations` tables are V2 compliant and exist.

- 🔴 Implement `food\_water\_source\_types\_master` Table: Execute the DDL for Version 1.2.1 as specified, ensuring role names in RLS policies are correct.

- 🔴 Apply Triggers: Ensure the `set\_current\_timestamp\_updated\_at` trigger and the `trigger\_cleanup\_food\_water\_source\_type\_translations` are created and correctly applied.

- 🔴 Initial Population / Seed Data: Insert the initial set of food/water source types using the seed data examples provided in the original `4.8 food\_water\_source\_types\_master.docx` (adjusting for `label` and `description` columns), and ensure `is\_active` is set to `true` and `created\_by\_profile\_id` is populated for these initial records.

- 🟠 RLS Helper Functions: Ensure that helper functions like `public.has\_role\_on\_profile(auth.uid(), 'role\_name')` are defined, secure, and used consistently in RLS policies.

- 🟠 RLS Policies: Implement and thoroughly test the RLS policies defined for this table.

- 🟢 Iconography Coordination: Ensure the `icon\_identifier` list used in seed data is coordinated with the UI/UX team for the preparation of corresponding visual assets.

- 🟢 Translation Entries: Prepare initial English entries for all translatable fields (`label`, `description`) in the `public.translations` table, ensuring they are correctly linked.