<https://gemini.google.com/u/1/app/9ed6b32c76b55d3a?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/8d62cfd96b565ac8>

<https://gemini.google.com/u/1/app/2d30425fc8a5984a>

\* \* \* \* \*

### Updated Production-Ready Specification

Table Name: `public.usage\_types\_master`

1\. Purpose & Primary Use-Cases

This table stores a canonical, centrally managed list of distinct permitted or common usage types for pilgrimage trails (e.g., "walking\_only," "cycling\_allowed\_sections," "equestrian\_friendly"). It provides structured, translatable data for consistent classification, clear communication to users, UI representation (icons, display names), and filtering capabilities across the platform.

Key User-Story Touchpoints:

- Pilgrim (Anna) - Story A1 (Route Exploration): Allows Anna to understand how a trail can be used and to filter trails based on usage types that match her intended activity.

- Platform Administrator (Admin Team) - Story D1 (Managing Content): Admins manage the definitive list of usage types, ensuring clarity and consistency.

- UI/UX: Provides structured data (`code`, `icon\_identifier`, and link to translations) for clear and consistent display.

2\. Schema

| Column Name | Data Type | Constraints | Description (Translatable fields via public.translations) |

| `id` | `integer` | PRIMARY KEY, GENERATED ALWAYS AS IDENTITY | Unique identifier for each usage type. |

| `code` | `text` | NOT NULL, UNIQUE, CHECK (`code` ~ '^[a-z0-9\_]+$' AND char\_length(`code`) &lt;= 50) | Machine-readable, stable code (e.g., "cycling\_allowed\_sections"). Not for direct display. |

| `name` | `text` | | \*Handled by `public.translations` table\* (English base for reference if desired, but primary source is `translations`). |

| `description` | `text` | | \*Handled by `public.translations` table\* (English base for reference if desired, but primary source is `translations`). |

| `icon\_identifier` | `text` | CHECK (`icon\_identifier` IS NULL OR char\_length(`icon\_identifier`) &lt;= 100) | Identifier (e.g., CSS class, SVG name) for an icon. |

| `notes` | `text` | | Internal administrative notes. Not typically for public display. |

| `display\_order` | `integer` | NOT NULL DEFAULT 0 | Optional order for displaying usage types in UI lists/filters. |

| `is\_active` | `boolean` | NOT NULL DEFAULT `true` | Flag to indicate if the usage type is active and available for use. |

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

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

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

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

3\. PostgreSQL DDL

SQL

```

-- This DDL assumes the 'public' schema is in use and 'public.profiles' table exists.

CREATE TABLE public.usage\_types\_master (

id INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY, --

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

icon\_identifier TEXT CHECK (icon\_identifier IS NULL OR char\_length(icon\_identifier) <= 100), --

notes TEXT, --

display\_order INTEGER NOT NULL DEFAULT 0, --

is\_active BOOLEAN NOT NULL DEFAULT true, -- V2 Enhancement

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(), --

created\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- V2 Enhancement

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(), --

updated\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL -- V2 Enhancement

);

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_usage\_types\_master\_code ON public.usage\_types\_master(code); --

CREATE INDEX IF NOT EXISTS idx\_usage\_types\_master\_display\_order ON public.usage\_types\_master(display\_order); --

CREATE INDEX IF NOT EXISTS idx\_usage\_types\_master\_is\_active ON public.usage\_types\_master(is\_active); -- V2 Enhancement

-- Trigger for audit columns (created\_by\_profile\_id, updated\_at, updated\_by\_profile\_id)

-- Reusing the generic master table audit trigger function defined for terrain\_types\_master

-- Ensure 'public.set\_master\_table\_audit\_meta()' is defined as:

-- CREATE OR REPLACE FUNCTION public.set\_master\_table\_audit\_meta()

-- RETURNS TRIGGER AS $$

-- BEGIN

-- NEW.updated\_at = NOW();

-- IF (TG\_OP = 'INSERT') THEN

-- IF NEW.created\_by\_profile\_id IS NULL THEN

-- NEW.created\_by\_profile\_id := auth.uid();

-- END IF;

-- IF NEW.updated\_by\_profile\_id IS NULL THEN

-- NEW.updated\_by\_profile\_id := auth.uid();

-- END IF;

-- ELSIF (TG\_OP = 'UPDATE') THEN

-- NEW.updated\_by\_profile\_id = auth.uid();

-- NEW.created\_at = OLD.created\_at;

-- NEW.created\_by\_profile\_id = OLD.created\_by\_profile\_id;

-- END IF;

-- RETURN NEW;

-- END;

-- $$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_usage\_types\_master\_audit\_meta

BEFORE INSERT OR UPDATE ON public.usage\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.set\_master\_table\_audit\_meta(); -- Using the same generic trigger

-- Comments

COMMENT ON TABLE public.usage\_types\_master IS 'Master list of distinct permitted or common usage types for trails (e.g., walking\_only, cycling\_allowed\_sections). Provides canonical definitions. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.id IS 'Unique identifier for the usage type. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.code IS 'A machine-readable, stable code for the usage type (e.g., "cycling\_allowed\_sections"). Used for internal referencing and linking, not for direct public display. Should be lowercase with underscores. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.icon\_identifier IS 'Identifier (e.g., CSS class name, SVG file name) for an icon representing this usage type. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.notes IS 'Internal administrative notes about this usage type, its definition, or intended application. Not typically for public display. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.display\_order IS 'Optional: an integer value to control the order in which usage types are presented in administrative interfaces or user-facing selection lists/filters. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.is\_active IS 'Flag to indicate if the usage type is active and available for new associations. Default true. Version: V2.';

COMMENT ON COLUMN public.usage\_types\_master.created\_at IS 'Timestamp of when the usage type record was created. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.created\_by\_profile\_id IS 'Profile ID of the user who created this usage type record. Version: V2.';

COMMENT ON COLUMN public.usage\_types\_master.updated\_at IS 'Timestamp of when the usage type record was last updated. Auto-updated by trigger. Version: V2.'; --

COMMENT ON COLUMN public.usage\_types\_master.updated\_by\_profile\_id IS 'Profile ID of the user who last updated this usage type record. Version: V2.';

```

4\. JSON Schema Mirror

JSON

```

{

"title": "usage\_type\_master",

"description": "Master list of distinct permitted or common usage types for trails. Version: V2.",

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int32",

"description": "Unique identifier for the usage type. Read-only.",

"readOnly": true

},

"code": {

"type": "string",

"maxLength": 50,

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

"description": "A machine-readable, stable code (e.g., \"cycling\_allowed\_sections\"). Not for direct public display. Must be unique."

},

"icon\_identifier": {

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

"maxLength": 100,

"description": "Identifier (e.g., CSS class name, file name) for an icon representing this usage type."

},

"notes": {

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

"description": "Internal administrative notes about this usage type."

},

"display\_order": {

"type": "integer",

"default": 0,

"description": "Optional: an order for displaying usage types in admin UIs or selection lists."

},

"is\_active": {

"type": "boolean",

"default": true,

"description": "Flag to indicate if the usage type is active and available for use."

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of record creation. Read-only.",

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who created this record. Read-only.",

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of last update. Read-only.",

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who last updated this record. Read-only.",

"readOnly": true

}

},

"required": [

"code",

"display\_order",

"is\_active",

"created\_at",

"updated\_at"

],

"primary\_key": ["id"],

"unique\_constraints": [

{"columns": ["code"], "name": "usage\_types\_master\_code\_key"}

]

}

```

5\. Relationships & Integrity

- This is a master data table.

- Referenced by `public.trail\_usage\_types` (junction table): `usage\_type\_id` FK references `usage\_types\_master.id` with `ON DELETE RESTRICT`. This prevents deletion of a master type if actively used.

6\. Multilingual Strategy

- The `code` field is language-neutral.

- User-facing display "name" and "description" are managed in the central `public.translations` table, keyed by `table\_name='usage\_types\_master'`, `column\_name` ('name' or 'description'), and `row\_id=usage\_types\_master.id`.

7\. Role-Based Workflow & RLS Notes

- Content Management: Foundational master data, typically managed by Platform Administrators.

- Workflow Fields: `created\_at`, `updated\_at`, `created\_by\_profile\_id`, `updated\_by\_profile\_id`, and `is\_active` manage lifecycle and audit.

- Note: The RLS policies outlined above rely on the existence and correct implementation of global RLS helper functions (e.g., public.has\_role(TEXT), public.is\_platform\_admin(), specific regional/trail management checks) that authenticate users and verify their roles stored in the public.profiles table." This reinforces that the table-specific RLS is part of a larger auth system.

- RLS Policy Stubs:

- Public Read Access: Publicly readable for active types.

SQL

```

CREATE POLICY "Allow public read access to active usage\_types\_master"

ON public.usage\_types\_master FOR SELECT

USING (is\_active = true); -- V2 Enhancement

```

- Admin Write Access: Restricted to privileged roles.

SQL

```

CREATE POLICY "Allow admin full access on usage\_types\_master"

ON public.usage\_types\_master FOR ALL

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

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

```

- Audit trigger `set\_master\_table\_audit\_meta` (or equivalent) is `SECURITY DEFINER`.

8\. ENUM vs. Lookup Discussion

- This table is a lookup table, replacing the old `trails.permitted\_uses TEXT[]` field.

- Advantages: Allows richer associated data (`icon\_identifier`, `notes`), easier i18n, and robust querying without schema changes for new types.

9\. UI/UX Enablement

- `code`: Stable internal identifier.

- `icon\_identifier`: Informs UI which icon to display.

- Translated names/descriptions from `public.translations` used for display.

- `display\_order`: Controls presentation order in UI.

- `is\_active`: Ensures only relevant types are shown in selection lists.

10\. Key Considerations & Definitions

- `code` Uniqueness and Stability: `code` must be unique and stable (lowercase snake\_case).

- Icon System Dependency: `icon\_identifier` relies on a frontend icon system.

- Granularity: Define types at a meaningful level.

11\. Scalability & Future-Proofing

- Expected to have a relatively small number of rows.

- Easily extensible by adding new rows or columns.

12\. Next-Action Checklist

1. 🔴 Implement DDL: Create the `public.usage\_types\_master` table using the updated DDL, including the new `is\_active`, `created\_by\_profile\_id`, `updated\_by\_profile\_id` columns, indexes, and the audit trigger (e.g., `set\_master\_table\_audit\_meta`).

2. 🔴 Implement Orphaned Translation Cleanup Trigger: Add an `AFTER DELETE` trigger on `public.usage\_types\_master` that calls `public.cleanup\_related\_translations('usage\_types\_master', OLD.id)` for 'name' and 'description' fields.

3. 🟠 Define and Populate Initial Seed Data: Define essential V1 seed rows for `usage\_types\_master` (e.g., `code`, `icon\_identifier`, `display\_order`, `is\_active`). Seed English `name` and `description` into `public.translations`.

4. 🟠 Implement RLS Policies & Helper Functions: Define, implement, and test RLS policies, including `public.is\_platform\_admin()`.

5. 🟠 Confirm Referencing Tables: Ensure `public.trail\_usage\_types.usage\_type\_id` correctly references this table with `ON DELETE RESTRICT`.

6. 🟢 Application Logic for `is\_active`: Ensure application logic (e.g., for populating admin UIs or selection dropdowns) filters for `is\_active = true` when fetching usage types for new associations.

\* \* \* \* \*