<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

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

\* \* \* \* \*

### Updated Production-Ready Specification

Table Name: `public.trail\_usage\_types`

1\. Purpose & Primary Use-Cases

This table establishes a many-to-many relationship between specific trails and the types of usage permitted or characteristic of them (e.g., walking, cycling on designated sections). It allows a single trail to be associated with multiple predefined usage types, enabling clear information display for users, supporting platform filtering capabilities, and facilitating content management by tracking these associations. This replaces the old trails.permitted\_uses text array.

Key User-Story Touchpoints:

- Pilgrim (Anna) - Story A1 (Route Exploration): Enables Anna to see a clear list of permitted uses for a trail (with translated names and icons via `usage\_types\_master`) and to filter trails based on activities she's interested in or restrictions she needs to observe.

- Application Logic/Filtering: Allows the application to fetch all permitted usage types for a given trail and enables users to filter or search for trails based on these usage types.

- Data Management (Admin/Content Manager): Provides the mechanism for authorized users to associate multiple predefined usage types with a trail entity and track when these associations were made and by whom.

2\. Schema

| Column Name | Data Type | Constraints | Description |

| `trail\_id` | `bigint` | PRIMARY KEY (Component), NOT NULL, REFERENCES `public.trails(id)` ON DELETE CASCADE | The ID of the trail. |

| `usage\_type\_id` | `integer` | PRIMARY KEY (Component), NOT NULL, REFERENCES `public.usage\_types\_master(id)` ON DELETE RESTRICT | The ID of the usage type (from `usage\_types\_master`) associated with the trail. |

| `created\_at` | `timestamptz` | NOT NULL DEFAULT `now()` | Timestamp indicating when this specific trail-usage type link was created. |

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

| `updated\_at` | `timestamptz` | NOT NULL DEFAULT `now()` | Timestamp indicating when this link record was last updated (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 link. |

3\. PostgreSQL DDL

SQL

```

-- This DDL assumes that 'public.trails' (with a BIGINT PK 'id'),

-- 'public.usage\_types\_master' (with an INTEGER PK 'id' and an 'is\_active' boolean column),

-- and 'public.profiles' (with a UUID PK 'id') tables already exist.

CREATE TABLE public.trail\_usage\_types (

trail\_id BIGINT NOT NULL REFERENCES public.trails(id) ON DELETE CASCADE, -- [cite: 955]

usage\_type\_id INTEGER NOT NULL REFERENCES public.usage\_types\_master(id) ON DELETE RESTRICT, -- [cite: 955]

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(), -- [cite: 955]

created\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 955]

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(), -- [cite: 955]

updated\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 955]

PRIMARY KEY (trail\_id, usage\_type\_id) -- [cite: 955]

);

-- Indexes:

-- The composite primary key will automatically create a unique index on (trail\_id, usage\_type\_id). [cite: 956]

-- Additional individual indexes can be useful depending on query patterns:

CREATE INDEX IF NOT EXISTS idx\_trail\_usage\_types\_trail\_id ON public.trail\_usage\_types(trail\_id); -- [cite: 957]

CREATE INDEX IF NOT EXISTS idx\_trail\_usage\_types\_usage\_type\_id ON public.trail\_usage\_types(usage\_type\_id); -- [cite: 958]

-- Trigger for created\_by\_profile\_id, updated\_at, and updated\_by\_profile\_id

CREATE OR REPLACE FUNCTION public.set\_trail\_usage\_type\_modification\_meta()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'INSERT') THEN

NEW.created\_by\_profile\_id = auth.uid(); -- [cite: 959]

NEW.updated\_at = NOW(); -- Ensures updated\_at is set on insert [cite: 959]

NEW.updated\_by\_profile\_id = auth.uid(); -- Ensures updated\_by\_profile\_id is set on insert [cite: 960]

ELSIF (TG\_OP = 'UPDATE') THEN

NEW.updated\_at = NOW(); -- [cite: 961]

NEW.updated\_by\_profile\_id = auth.uid(); -- [cite: 961]

-- Keep OLD.created\_at and OLD.created\_by\_profile\_id

NEW.created\_at = OLD.created\_at; -- [cite: 961]

NEW.created\_by\_profile\_id = OLD.created\_by\_profile\_id; -- [cite: 961]

END IF;

RETURN NEW; -- [cite: 962]

END;

$$ LANGUAGE plpgsql SECURITY DEFINER; -- [cite: 962]

CREATE TRIGGER trigger\_trail\_usage\_types\_modification\_meta

BEFORE INSERT OR UPDATE ON public.trail\_usage\_types

FOR EACH ROW

EXECUTE FUNCTION public.set\_trail\_usage\_type\_modification\_meta(); -- [cite: 963]

-- Comments:

COMMENT ON TABLE public.trail\_usage\_types IS 'Junction table linking trails to their permitted usage types (many-to-many). Replaces the old trails.permitted\_uses text array. Version: V2.'; -- [cite: 963, 964]

COMMENT ON COLUMN public.trail\_usage\_types.trail\_id IS 'Foreign key to the public.trails table. Version: V2.'; -- [cite: 965]

COMMENT ON COLUMN public.trail\_usage\_types.usage\_type\_id IS 'Foreign key to the public.usage\_types\_master table. Deletion of a master usage type is restricted if in use. Version: V2.'; -- [cite: 966]

COMMENT ON COLUMN public.trail\_usage\_types.created\_at IS 'Timestamp when the trail-usage type association was made. Version: V2.'; -- [cite: 967]

COMMENT ON COLUMN public.trail\_usage\_types.created\_by\_profile\_id IS 'Profile ID (public.profiles.id) of the user who initially created this trail-usage type association. Version: V2.'; -- [cite: 968]

COMMENT ON COLUMN public.trail\_usage\_types.updated\_at IS 'Timestamp when the trail-usage type association was last updated. Auto-updated by trigger. Version: V2.'; -- [cite: 969]

COMMENT ON COLUMN public.trail\_usage\_types.updated\_by\_profile\_id IS 'Profile ID (public.profiles.id) of the user who last updated this trail-usage type association. Version: V2.'; -- [cite: 970]

```

4\. JSON Schema Mirror

JSON

```

{

"title": "trail\_usage\_type\_link",

"description": "Links trails to their permitted usage types (many-to-many). This replaces the old text array for permitted uses on a trail. Version: V2.",

"type": "object",

"properties": {

"trail\_id": {

"type": "integer",

"format": "int64",

"description": "Foreign key referencing the ID of the trail (public.trails.id)." -- [cite: 972]

},

"usage\_type\_id": {

"type": "integer",

"format": "int32",

"description": "Foreign key referencing the ID of the usage type from public.usage\_types\_master." -- [cite: 973]

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of when this link record was created. Read-only.", -- [cite: 974]

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who created this link. Read-only.", -- [cite: 974]

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of when this link record was last updated. Read-only.", -- [cite: 974]

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who last updated this link. Read-only.", -- [cite: 974]

"readOnly": true

}

},

"required": [

"trail\_id",

"usage\_type\_id",

"created\_at",

"updated\_at"

],

"primary\_key": ["trail\_id", "usage\_type\_id"] -- [cite: 975]

}

```

5\. Relationships & Integrity

- Junction Table: This is a pure junction table establishing the many-to-many relationship between `public.trails` and `public.usage\_types\_master`.

- Foreign Keys & `ON DELETE` Actions:

- `trail\_id` references `public.trails(id)`: `ON DELETE CASCADE`. If a trail is deleted, its associated usage type links are automatically removed.

- `usage\_type\_id` references `public.usage\_types\_master(id)`: `ON DELETE RESTRICT`. This protects master data integrity by preventing deletion of a usage type if it's linked to any trail.

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

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

- Primary Key: Composite `(trail\_id, usage\_type\_id)` ensures a trail cannot be linked to the same usage type multiple times.

- Mermaid ER Diagram Snippet:

Code snippet

```

erDiagram

trails {

bigint id PK

text name

}

usage\_types\_master {

integer id PK

text code UK

boolean is\_active -- V2 addition to master

}

profiles {

uuid id PK

}

trail\_usage\_types {

bigint trail\_id PK FK --

integer usage\_type\_id PK FK --

timestamptz created\_at --

uuid created\_by\_profile\_id FK --

timestamptz updated\_at --

uuid updated\_by\_profile\_id FK --

}

trails ||--|{ trail\_usage\_types : "has\_usage\_type (CASCADE)" --

usage\_types\_master ||--|{ trail\_usage\_types : "applies\_to\_trail (RESTRICT)" --

trail\_usage\_types }o--|| profiles : "created\_by (SET NULL)" --

trail\_usage\_types }o--|| profiles : "updated\_by (SET NULL)" --

```

6\. Multilingual Strategy

This table itself contains no directly translatable text. It relies on public.usage\_types\_master for the definition of usage types. User-facing names and descriptions associated with usage\_types\_master.id are translated via the central public.translations table.

7\. Role-Based Workflow & RLS Notes

- Workflow Fields: `created\_at`, `created\_by\_profile\_id`, `updated\_at`, `updated\_by\_profile\_id` provide a complete audit trail for link creation and modification.

- Content Management: Links are typically managed by Platform Administrators or Regional Content Managers.

- 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 (Conceptual):

- Public Read Access: Readable if the associated trail is public.

SQL

```

CREATE POLICY "Allow public read access to trail\_usage\_types"

ON public.trail\_usage\_types FOR SELECT

USING (

EXISTS (

SELECT 1 FROM public.trails t

WHERE t.id = trail\_usage\_types.trail\_id

AND t.content\_visibility\_status = 'published' -- Or your equivalent public status [cite: 995]

AND t.deleted\_at IS NULL

)

AND EXISTS ( -- Ensure linked usage type is active

SELECT 1 FROM public.usage\_types\_master utm

WHERE utm.id = trail\_usage\_types.usage\_type\_id

AND utm.is\_active = true

)

);

```

- Admin/Manager Write Access: Users authorized to edit a trail should be able to manage its usage type associations.

SQL

```

CREATE POLICY "Allow authorized users to manage trail\_usage\_types"

ON public.trail\_usage\_types FOR ALL -- Covers INSERT, UPDATE, DELETE [cite: 997]

USING (

public.check\_user\_can\_edit\_trail(trail\_usage\_types.trail\_id) -- Placeholder [cite: 997]

)

WITH CHECK (

public.check\_user\_can\_edit\_trail(trail\_usage\_types.trail\_id) -- Placeholder [cite: 997]

AND EXISTS ( -- Ensure associating with an active usage type

SELECT 1 FROM public.usage\_types\_master utm

WHERE utm.id = trail\_usage\_types.usage\_type\_id

AND utm.is\_active = true

)

);

```

- Audit trigger `set\_trail\_usage\_type\_modification\_meta` is `SECURITY DEFINER`.

8\. ENUM vs. Lookup Discussion

This junction table supports the use of public.usage\_types\_master (a lookup table), which is appropriate for managing usage types.

9\. UI/UX Enablement

- Displays & Information: Allows UI to display characteristic usage types for a trail by joining `trails` → `trail\_usage\_types` → `usage\_types\_master`. Icons are driven by `usage\_types\_master.icon\_identifier`, and names/descriptions via `public.translations`.

- Filtering: Enables backend filtering for trails with specific usage types.

- Audit Trails (Admin UI): Audit columns provide visibility into changes.

10\. Key Considerations & Definitions

- Purpose: Defines trail-level permitted usage characteristics.

- Data Integrity:

- Composite PK `(trail\_id, usage\_type\_id)` prevents duplicate links.

- `ON DELETE RESTRICT` on `usage\_type\_id` FK is crucial for protecting `usage\_types\_master` data.

- Auditability: Full audit trail for link creation/modification supported.

- Application Logic: When creating new links, the application should ideally only allow selection of `usage\_types\_master` records where `is\_active = true`.

11\. Scalability & Future-Proofing

- Standard junction table design scales well.

- Audit columns improve maintainability.

- Could be expanded with attributes specific to the trail-usage link if needed. For V1, it remains a pure link with audit information.

12\. Next-Action Checklist

1. 🔴 Implement DDL: Create the `public.trail\_usage\_types` table using the updated DDL, ensuring all constraints (PK, FKs with correct `ON DELETE` actions), indexes, and the `set\_trail\_usage\_type\_modification\_meta` trigger are correctly defined and attached.

2. 🟠 Verify Referenced Table PK Data Types: Confirm `public.trails.id` is `BIGINT` and `public.usage\_types\_master.id` is `INTEGER`.

3. 🟠 Implement RLS Policies & Helper Functions: Define, implement, and thoroughly test RLS policies, including the `public.check\_user\_can\_edit\_trail(BIGINT)` helper function and checks for `usage\_types\_master.is\_active` in write policies.

4. 🟠 Data Migration (if applicable): If migrating from an old `trails.permitted\_uses TEXT[]`, plan and execute migration. Populate audit columns for migrated records.

5. 🟢 Review Application Logic: Update application code to query via this junction table. Ensure logic for creating new links presents only active usage types for selection.

\* \* \* \* \*