https://gemini.google.com/u/1/app/431cc17871bb821b

```

# Media Linking & Processing Specifications (V1 Draft)

Date: May 18, 2025

## 📜 Part 1: Table Specification - `public.trail\_media`

### 1. Purpose & Primary Use-Cases

This table links records from `public.trails` to specific media items in `public.media`, assigning a semantic role to each link. It enables trails to have multiple associated images or other media files, such as a prominent hero image, a gallery of representative photos, and illustrative static map graphics. This allows for rich visual storytelling and detailed information presentation for each trail.

\*\*Key User-Story Touchpoints:\*\*

\* Pilgrim (Anna): Views hero images when exploring a trail, browses image galleries to get a feel for the trail's scenery and points of interest, and looks at static map images for overviews.

\* Regional Content Manager (Sofia) / Platform Administrator: Uploads, selects, and curates images for trail hero sections, galleries, and map illustrations. Manages captions, alt text, and display order for these media items.

\* System/Developer: Retrieves media based on roles for display in different UI contexts (e.g., fetching the `featured\_image` for the trail page header, or all `gallery\_image` items for a carousel).

### 2. Schema

| Column Name | Data Type | Constraints | Description (Translatable fields marked) |

| :---------------------- | :------------ | :---------------------------------------------------------------------------------------------------------------- | :--------------------------------------------------------------------------------------------------------------------------------------------------------------------- |

| `id` | `BIGINT` | `PRIMARY KEY, GENERATED ALWAYS AS IDENTITY` | Unique identifier for this trail-media link. |

| `trail\_id` | `BIGINT` | `NOT NULL, REFERENCES public.trails(id) ON DELETE CASCADE` | The ID of the trail this media is associated with. |

| `media\_id` | `UUID` | `NOT NULL, REFERENCES public.media(id) ON DELETE RESTRICT` | The ID of the media item from `public.media`. `ON DELETE RESTRICT` to prevent deleting media still in use by a trail. |

| `media\_role\_code` | `TEXT` | `NOT NULL, REFERENCES public.media\_roles\_master(code) ON DELETE RESTRICT ON UPDATE CASCADE` | Semantic role of the media in the context of this trail (e.g., 'featured\_image', 'gallery\_image', 'map\_image\_static'). From `public.media\_roles\_master`. |

| `display\_order` | `SMALLINT` | `NOT NULL, DEFAULT 0` | Order for displaying media items within a specific role for this trail (e.g., sequence in a gallery). |

| `caption\_override` | `TEXT` | `NULL` | Optional override for the media's default caption, specific to its use with this trail. Primary reference language (English). (Translatable via `public.translations`) |

| `alt\_text\_override` | `TEXT` | `NULL` | Optional override for the media's default alt text, specific to its use with this trail, for accessibility. Primary reference language (English). (Translatable via `public.translations`) |

| `created\_at` | `TIMESTAMPTZ` | `NOT NULL, DEFAULT now()` | Timestamp of when this link record was created. |

| `created\_by\_profile\_id` | `UUID` | `NULL, 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 of when this link record was last updated (auto-updated by trigger). |

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

| | | `UNIQUE (trail\_id, media\_id, media\_role\_code)` | Prevents linking the same media item with the same role to the same trail multiple times. |

| | | `UNIQUE (trail\_id, display\_order, media\_role\_code)` | Ensures display order is unique within a specific role for a given trail. |

### 3. PostgreSQL DDL

```sql

-- Assumes the following tables exist:

-- public.trails (id BIGINT PK)

-- public.media (id UUID PK)

-- public.media\_roles\_master (code TEXT PK)

-- public.profiles (id UUID PK)

-- public.translations (for translatable fields)

-- Assumes standard functions like public.set\_current\_timestamp\_updated\_at() or extensions.moddatetime exist.

-- Assumes public.cleanup\_related\_translations(TEXT, TEXT) function exists.

CREATE TABLE public.trail\_media (

id BIGINT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

trail\_id BIGINT NOT NULL REFERENCES public.trails(id) ON DELETE CASCADE,

media\_id UUID NOT NULL REFERENCES public.media(id) ON DELETE RESTRICT,

media\_role\_code TEXT NOT NULL REFERENCES public.media\_roles\_master(code) ON DELETE RESTRICT ON UPDATE CASCADE,

display\_order SMALLINT NOT NULL DEFAULT 0,

caption\_override TEXT NULL,

alt\_text\_override TEXT NULL,

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

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

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

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

CONSTRAINT uq\_trail\_media\_link UNIQUE (trail\_id, media\_id, media\_role\_code),

CONSTRAINT uq\_trail\_media\_role\_order UNIQUE (trail\_id, display\_order, media\_role\_code)

);

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_trail\_media\_trail\_id ON public.trail\_media(trail\_id);

CREATE INDEX IF NOT EXISTS idx\_trail\_media\_media\_id ON public.trail\_media(media\_id);

CREATE INDEX IF NOT EXISTS idx\_trail\_media\_media\_role\_code ON public.trail\_media(media\_role\_code);

CREATE INDEX IF NOT EXISTS idx\_trail\_media\_created\_by\_profile\_id ON public.trail\_media(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_trail\_media\_updated\_by\_profile\_id ON public.trail\_media(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

-- Comments

COMMENT ON TABLE public.trail\_media IS 'Links trails to media items (galleries, hero images, static maps), defining display order, role, and translatable caption/alt-text overrides. V1.0';

COMMENT ON COLUMN public.trail\_media.id IS 'Surrogate unique identifier for the trail-media link.';

COMMENT ON COLUMN public.trail\_media.trail\_id IS 'Foreign key to the public.trails table.';

COMMENT ON COLUMN public.trail\_media.media\_id IS 'Foreign key to the public.media table. Restricted delete to prevent orphaned links if media is in use.';

COMMENT ON COLUMN public.trail\_media.media\_role\_code IS 'Semantic role of the media in relation to the trail. FK to public.media\_roles\_master.code.';

COMMENT ON COLUMN public.trail\_media.display\_order IS 'Order for displaying media items within a specific role for this trail (e.g., sequence in a gallery). Default 0.';

COMMENT ON COLUMN public.trail\_media.caption\_override IS 'Optional override for the media''s default caption, specific to its use with this trail. Primary reference language (English). (Translatable via public.translations using trail\_media.id).';

COMMENT ON COLUMN public.trail\_media.alt\_text\_override IS 'Optional override for the media''s default alt text, specific to its use with this trail. Primary reference language (English). (Translatable via public.translations using trail\_media.id).';

COMMENT ON COLUMN public.trail\_media.created\_at IS 'Timestamp of when this link record was created.';

COMMENT ON COLUMN public.trail\_media.created\_by\_profile\_id IS 'Profile ID of the user who created this link.';

COMMENT ON COLUMN public.trail\_media.updated\_at IS 'Timestamp of when this link record was last updated (auto-updated by trigger).';

COMMENT ON COLUMN public.trail\_media.updated\_by\_profile\_id IS 'Profile ID of the user who last updated this link.';

-- Standard updated\_at trigger

CREATE TRIGGER trigger\_trail\_media\_set\_updated\_at

BEFORE UPDATE ON public.trail\_media

FOR EACH ROW

EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at(); -- Or extensions.moddatetime('updated\_at')

COMMENT ON TRIGGER trigger\_trail\_media\_set\_updated\_at ON public.trail\_media IS 'Automatically updates the updated\_at timestamp on row modification.';

-- Orphaned translation cleanup trigger for trail\_media's own translatable fields

CREATE OR REPLACE FUNCTION public.cleanup\_trail\_media\_translations()

RETURNS TRIGGER AS $$

BEGIN

IF TG\_OP = 'DELETE' THEN

DELETE FROM public.translations

WHERE table\_identifier = 'trail\_media'

AND row\_foreign\_key = OLD.id::TEXT; -- Uses the surrogate PK of trail\_media

END IF;

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

COMMENT ON FUNCTION public.cleanup\_trail\_media\_translations() IS 'Removes related entries from public.translations for trail\_media.caption\_override and trail\_media.alt\_text\_override when a trail\_media record is deleted. Runs as SECURITY DEFINER.';

CREATE TRIGGER trigger\_cleanup\_trail\_media\_translations\_after\_delete

AFTER DELETE ON public.trail\_media

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_trail\_media\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_trail\_media\_translations\_after\_delete ON public.trail\_media IS 'After a trail\_media link is deleted, removes its associated translations from the public.translations table.';

-- Conceptual Trigger to update media.last\_linked\_or\_used\_at

-- This function would be global, and each media linking table (like trail\_media)

-- would call it.

-- CREATE OR REPLACE FUNCTION public.update\_media\_last\_linked\_timestamp\_on\_link()

-- RETURNS TRIGGER AS $$

-- BEGIN

-- UPDATE public.media

-- SET last\_linked\_or\_used\_at = NOW()

-- WHERE id = NEW.media\_id;

-- RETURN NEW;

-- END;

-- $$ LANGUAGE plpgsql SECURITY DEFINER;

-- CREATE TRIGGER trigger\_trail\_media\_update\_media\_timestamp

-- AFTER INSERT OR UPDATE OF media\_id ON public.trail\_media

-- FOR EACH ROW

-- EXECUTE FUNCTION public.update\_media\_last\_linked\_timestamp\_on\_link();

-- COMMENT ON TRIGGER trigger\_trail\_media\_update\_media\_timestamp ON public.trail\_media

-- IS 'Updates media.last\_linked\_or\_used\_at when a media item is linked or its link is updated in trail\_media.';

-- (A similar trigger might be needed on DELETE to check if media should be "unlinked", though this is complex if media is shared)

```

### 4\. JSON Schema Mirror

JSON

```

{

"title": "trail\_media\_link",

"description": "Links trails to media items (galleries, hero images, static maps), defining display order, role, and translatable caption/alt-text overrides. V1.0",

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64",

"description": "Surrogate unique identifier for the trail-media link.",

"readOnly": true

},

"trail\_id": {

"type": "integer",

"format": "int64",

"description": "Foreign key to the public.trails table."

},

"media\_id": {

"type": "string",

"format": "uuid",

"description": "Foreign key to the public.media table."

},

"media\_role\_code": {

"type": "string",

"description": "Semantic role of the media in relation to the trail. FK to public.media\_roles\_master.code."

},

"display\_order": {

"type": "integer",

"format": "int16",

"default": 0,

"description": "Order for displaying media items within a specific role for this trail."

},

"caption\_override": {

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

"description": "Optional override for the media's default caption, specific to this trail link. Primary reference language (English). (Translatable via public.translations using trail\_media.id)."

},

"alt\_text\_override": {

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

"description": "Optional override for the media's default alt text, specific to this trail link. Primary reference language (English). (Translatable via public.translations using trail\_media.id)."

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of when this link record was created.",

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who created this link."

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of when this link record was last updated.",

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

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

}

},

"required": [

"trail\_id",

"media\_id",

"media\_role\_code",

"display\_order",

"created\_at",

"updated\_at"

],

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

"unique\_constraints": [

{"columns": ["trail\_id", "media\_id", "media\_role\_code"], "name": "uq\_trail\_media\_link"},

{"columns": ["trail\_id", "display\_order", "media\_role\_code"], "name": "uq\_trail\_media\_role\_order"}

],

"foreign\_keys": [

{"columns": ["trail\_id"], "references\_table": "public.trails", "references\_columns": ["id"], "on\_delete": "CASCADE"},

{"columns": ["media\_id"], "references\_table": "public.media", "references\_columns": ["id"], "on\_delete": "RESTRICT"},

{"columns": ["media\_role\_code"], "references\_table": "public.media\_roles\_master", "references\_columns": ["code"], "on\_delete": "RESTRICT", "on\_update": "CASCADE"},

{"columns": ["created\_by\_profile\_id"], "references\_table": "public.profiles", "references\_columns": ["id"], "on\_delete": "SET NULL"},

{"columns": ["updated\_by\_profile\_id"], "references\_table": "public.profiles", "references\_columns": ["id"], "on\_delete": "SET NULL"}

]

}

```

### 5\. Relationships & Integrity

- Primary Key: `id BIGINT` (surrogate key).

- Foreign Keys:

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

- `media\_id` references `public.media(id)` `ON DELETE RESTRICT`. Prevents deletion of media if it's actively linked to a trail.

- `media\_role\_code` references `public.media\_roles\_master(code)` `ON DELETE RESTRICT ON UPDATE CASCADE`. Ensures role integrity.

- Audit FKs (`created\_by\_profile\_id`, `updated\_by\_profile\_id`) reference `public.profiles(id)` `ON DELETE SET NULL`.

- Unique Constraints:

- `UNIQUE (trail\_id, media\_id, media\_role\_code)`: Prevents the same media item from being linked to the same trail with the exact same role multiple times.

- `UNIQUE (trail\_id, display\_order, media\_role\_code)`: Ensures `display\_order` is unique within a specific role for a given trail (e.g., for gallery image sequencing).

### 6\. Multilingual Strategy

- Translatable Fields: `caption\_override`, `alt\_text\_override`.

- Mechanism: These fields store their primary reference language (English) text directly. Translations into other languages are managed in the central `public.translations` table, linked by `table\_identifier = 'trail\_media'`, `column\_identifier` (e.g., 'caption\_override'), and `row\_foreign\_key = trail\_media.id::TEXT`.

- Fallback: If an override is `NULL`, the application should fall back to `public.media.default\_caption` or `public.media.default\_alt\_text` respectively.

- Orphan Cleanup: The `trigger\_cleanup\_trail\_media\_translations\_after\_delete` trigger ensures that when a `trail\_media` record is deleted, its specific translations for overrides are removed from `public.translations`.

### 7\. Role-Based Workflow & RLS Notes

- Content Management: Admins (`admin\_platform`) and Regional Content Managers (`regional\_content\_manager`) will typically manage these media links.

- RLS Policy Sketch:

- Public Read Access: `SELECT` allowed if the parent `public.trails` record is published and not deleted.

SQL

```

CREATE POLICY "Allow public read access to trail\_media"

ON public.trail\_media FOR SELECT

USING (

EXISTS (

SELECT 1 FROM public.trails t

WHERE t.id = trail\_media.trail\_id

AND t.content\_visibility\_status = 'published'::public.content\_visibility\_status\_enum -- Assuming ENUM value

AND t.deleted\_at IS NULL

)

);

```

- Content Manager Write Access: `INSERT`, `UPDATE`, `DELETE` allowed for roles like `admin\_platform` or `regional\_content\_manager` if they have rights to manage the parent `trail\_id`. This often involves a helper function.

SQL

```

CREATE POLICY "Allow content managers to manage trail\_media"

ON public.trail\_media FOR ALL

USING (

(SELECT public.has\_role('admin\_platform')) OR -- Using platform standard helper

(SELECT public.has\_role('regional\_content\_manager') AND public.user\_manages\_trail(trail\_media.trail\_id)) -- Example helper

)

WITH CHECK (

(SELECT public.has\_role('admin\_platform')) OR

(SELECT public.has\_role('regional\_content\_manager') AND public.user\_manages\_trail(trail\_media.trail\_id))

);

```

- (The `public.user\_manages\_trail(BIGINT)` helper function would need to be defined, checking regional assignments or other criteria).

### 8\. ENUM vs. Lookup Discussion

- N/A for this table directly, as `media\_role\_code` correctly references the `public.media\_roles\_master` lookup table.

### 9\. UI/UX Enablement

- Drives trail galleries (`gallery\_image` role) with ordering via `display\_order`.

- Provides the main hero image (`featured\_image` role) for trail pages.

- Allows for specific map graphics (`map\_image\_static` role).

- Contextual captions and alt text enhance accessibility and information.

### 10\. Key Considerations & Definitions

- Role Cardinality:

- A trail should likely have only one `featured\_image`. This would be an application-level or admin UI guideline, or potentially a more complex database constraint (e.g., filtered unique index on `(trail\_id, media\_role\_code) WHERE media\_role\_code = 'featured\_image'`). For V1, UI guidance is often sufficient.

- Multiple `gallery\_image` and `map\_image\_static` entries per trail are expected.

- `display\_order` Management: Admins/managers should be able to set and reorder. For new gallery images, the application could default to `MAX(display\_order) + 1` for that trail and role.

- `media\_id` ON DELETE RESTRICT: This is important. If a media item is deleted from `public.media`, but it's still linked here, the delete should be prevented to maintain integrity. An admin would need to unlink it from `trail\_media` first.

### 11\. Scalability & Future-Proofing

- Standard linking table structure; scales with the number of trails and media items.

- Use of `media\_roles\_master` allows for adding new types of media associations for trails in the future without schema changes to `trail\_media` itself (just new role codes).

### 12\. Seed Data

- N/A. This is a transactional linking table; data is created as media is associated with trails.

### 13\. Next-Action Checklist

1. 🔴 Verify Dependencies: Ensure `public.trails`, `public.media`, `public.media\_roles\_master`, `public.profiles`, and `public.translations` tables are created and V2 compliant.

2. 🔴 Define `media\_roles\_master` Entries: Confirm/add necessary `code` values to `public.media\_roles\_master` (e.g., `featured\_image`, `gallery\_image`, `map\_image\_static`).

3. 🔴 Implement DDL: Execute the `CREATE TABLE public.trail\_media` DDL, including all constraints, indexes, and comments.

4. 🔴 Implement Triggers:

- Apply the `updated\_at` trigger.

- Create and apply the `trigger\_cleanup\_trail\_media\_translations\_after\_delete`.

- \*(Optional but Recommended)\* Implement logic/triggers for `media.last\_linked\_or\_used\_at` updates via a global function called by this table's trigger.

5. 🟠 RLS Policies & Helper Functions: Define and implement RLS policies. This includes creating and testing any necessary helper functions like `public.user\_manages\_trail(BIGINT)`.

6. 🟢 Application Logic:

- Develop UI/backend logic for managing `display\_order`.

- Implement fallback logic for `caption\_override`/`alt\_text\_override` to use `media.default\_caption`/`alt\_text`.

- Consider application-level enforcement for role cardinalities if not done in DB (e.g., one `featured\_image`).

7. 🟢 Data Migration: If there's existing implicit trail media data, plan migration.

\* \* \* \* \*