## [**https://gemini.google.com/u/1/app/fd47ede3d905a3f2**](https://gemini.google.com/u/1/app/fd47ede3d905a3f2)

https://gemini.google.com/u/1/app/440838222ac514d7

Production-Ready Specification: public.media Table

Version: 2.2 (Checklist V2-05-17-B Applied)

Date: 2025-05-17

1. Purpose & Primary Use-Cases

Mission Statement: To act as the central metadata repository for every individual media file (images primarily for V1, extensible to other types) used on the platform, referencing files in Supabase Storage and holding descriptive, technical, workflow, and administrative information, including audit trails for creation and updates.

Key User-Story Touchpoints:

• Pilgrim (Anna): Views images associated with platform content.

• Host (Marco): Uploads and manages images for his accommodation.

• Regional Manager (Sofia): Manages media for regional content.

• Platform Administrator: Oversees all media, manages statuses, with auditable actions.

• System/Developer: Implements upload/optimization pipeline, serves correct image variants, relies on audit fields.

2. Schema

| column | data\_type | constraints | description |

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

| id | uuid | Primary Key, Default `gen\_random\_uuid()` | Unique identifier for the media item. |

| uploader\_profile\_id | uuid | Nullable, References `profiles(id)` ON DELETE SET NULL | Profile ID of the user who uploaded this media. (Serves as `created\_by`). If uploader's profile is deleted, media is kept, uploader link becomes NULL. |

| storage\_bucket\_name | text | Not Null | Name of the Supabase Storage bucket (e.g., 'platform-media'). |

| storage\_object\_path\_original | text | Not Null, Unique (`uq\_media\_storage\_object\_path`) | Full, unique path to the original uploaded object within its bucket (e.g., `uploads/images/2025/05/{uuid}\_original.jpg`). |

| file\_name\_original | text | Not Null | Original filename as provided by the uploader (e.g., "La Verna Sunset.jpg"). |

| file\_mime\_type | text | Not Null | MIME type of the original file (e.g., 'image/jpeg', 'image/png', 'application/pdf'). |

| file\_size\_bytes\_original | bigint | Nullable, CHECK (`file\_size\_bytes\_original` IS NULL OR `file\_size\_bytes\_original` > 0) | Size of the original uploaded file in bytes. |

| image\_width\_px\_original | integer | Nullable, CHECK (`image\_width\_px\_original` IS NULL OR `image\_width\_px\_original` > 0) | For images, the original width in pixels. Null if not an image or if not applicable. |

| image\_height\_px\_original | integer | Nullable, CHECK (`image\_height\_px\_original` IS NULL OR `image\_height\_px\_original` > 0) | For images, the original height in pixels. Null if not an image or if not applicable. |

| image\_variants\_json | jsonb | Nullable | 🟠 JSONB storing relative paths for optimized image versions. Populated by app layer/backend. Ex: `{"thumb\_s": "path/to\_thumb.webp", "display\_l": "path/to\_display.jpg"}`. |

| default\_alt\_text | text | Nullable | Default alternative text in the primary reference language for accessibility (WCAG). (Translatable via `public.translations`) |

| default\_caption | text | Nullable | Default caption in the primary reference language for display. (Translatable via `public.translations`) |

| attribution\_text | text | Nullable | Credit for the photographer/source. (Potentially Translatable via `public.translations`) |

| attribution\_url | text | Nullable, CHECK (`attribution\_url` IS NULL OR `attribution\_url` ~\* '^https?://.+') | URL linking to the source or license. Validated format. |

| licence | media\_licence\_enum | Nullable, Default `'all\_rights\_reserved'` | Usage license associated with the media item. |

| media\_asset\_type | media\_asset\_type\_enum | Not Null, Default `'image'` | General type of media asset (e.g., 'image', 'document\_pdf'). |

| media\_status | media\_status\_enum | Not Null, Default `'pending\_review'` | Current workflow/visibility status of the media item. |

| tags | text[] | Nullable | Array of keywords/tags for categorizing or searching media. |

| checksum\_sha256\_original | text | Nullable, Unique (`uq\_media\_checksum\_original`) | SHA256 checksum of the original file. Used for de-duplication. Unique if not NULL. |

| dominant\_color\_hex | text | Nullable, CHECK (`dominant\_color\_hex` IS NULL OR `dominant\_color\_hex` ~ '^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$') | Optional dominant color of an image (hex code), for UI placeholders. |

| last\_linked\_or\_used\_at | timestamp with time zone | Nullable | Timestamp when this media item was last actively linked. Useful for cleanup of orphaned media. Updated by triggers on linking tables. |

| created\_at | timestamp with time zone | Not Null, Default `now()` | Timestamp of when this media record was created. |

| updated\_at | timestamp with time zone | Not Null, Default `now()` | Timestamp of when this media record was last updated (auto-updated by general row update trigger). |

| updated\_by\_profile\_id | uuid | Nullable, References `public.profiles(id)` ON DELETE SET NULL | Profile ID of the user who last significantly updated this media record (e.g., status change, key metadata update). ON DELETE SET NULL. |

| deleted\_at | timestamp with time zone | Nullable | Timestamp for soft deletion. Media marked as deleted might be scheduled for actual file removal from storage. |

3. PostgreSQL DDL

-- ENUM Types (ensure these are created globally or before this table if not already existing)

DO $$

BEGIN

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_asset\_type\_enum') THEN

CREATE TYPE public.media\_asset\_type\_enum AS ENUM ('image', 'document\_pdf', 'audio\_clip', 'video\_clip', 'gpx\_file', 'other\_file');

END IF;

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_licence\_enum') THEN

CREATE TYPE public.media\_licence\_enum AS ENUM ('all\_rights\_reserved', 'cc\_by', 'cc\_by\_sa', 'cc\_by\_nd', 'cc\_by\_nc', 'cc\_by\_nc\_sa', 'cc\_by\_nc\_nd', 'cc0\_public\_domain', 'uploader\_owns\_contact\_for\_use', 'official\_permission\_granted', 'unknown\_licence');

END IF;

IF NOT EXISTS (SELECT 1 FROM pg\_type WHERE typname = 'media\_status\_enum') THEN

CREATE TYPE public.media\_status\_enum AS ENUM ('processing\_upload', 'pending\_review', 'published\_approved', 'rejected\_hidden', 'archived', 'error\_uploading');

END IF;

END$$;

-- Table Definition

CREATE TABLE public.media (

id uuid NOT NULL DEFAULT gen\_random\_uuid() PRIMARY KEY,

uploader\_profile\_id uuid NULL REFERENCES public.profiles(id) ON DELETE SET NULL, -- Serves as created\_by\_profile\_id

storage\_bucket\_name text NOT NULL,

storage\_object\_path\_original text NOT NULL,

file\_name\_original text NOT NULL,

file\_mime\_type text NOT NULL,

file\_size\_bytes\_original bigint NULL,

image\_width\_px\_original integer NULL,

image\_height\_px\_original integer NULL,

image\_variants\_json jsonb NULL,

default\_alt\_text text NULL,

default\_caption text NULL,

attribution\_text text NULL,

attribution\_url text NULL,

licence public.media\_licence\_enum NULL DEFAULT 'all\_rights\_reserved',

media\_asset\_type public.media\_asset\_type\_enum NOT NULL DEFAULT 'image',

media\_status public.media\_status\_enum NOT NULL DEFAULT 'pending\_review',

tags text[] NULL,

checksum\_sha256\_original text NULL,

dominant\_color\_hex text NULL,

last\_linked\_or\_used\_at timestamp with time zone NULL,

created\_at timestamp with time zone NOT NULL DEFAULT now(),

updated\_at timestamp with time zone NOT NULL DEFAULT now(),

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

deleted\_at timestamp with time zone NULL,

CONSTRAINT uq\_media\_storage\_object\_path UNIQUE (storage\_object\_path\_original),

CONSTRAINT uq\_media\_checksum\_original UNIQUE (checksum\_sha256\_original),

CONSTRAINT check\_file\_size\_positive CHECK (file\_size\_bytes\_original IS NULL OR file\_size\_bytes\_original > 0),

CONSTRAINT check\_image\_width\_positive CHECK (image\_width\_px\_original IS NULL OR image\_width\_px\_original > 0),

CONSTRAINT check\_image\_height\_positive CHECK (image\_height\_px\_original IS NULL OR image\_height\_px\_original > 0),

CONSTRAINT check\_dominant\_color\_hex\_format CHECK (dominant\_color\_hex IS NULL OR dominant\_color\_hex ~ '^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$'),

CONSTRAINT check\_attribution\_url\_format CHECK (attribution\_url IS NULL OR attribution\_url ~\* '^https?://.+')

);

-- Indexes

CREATE INDEX idx\_media\_uploader\_profile\_id ON public.media (uploader\_profile\_id) WHERE uploader\_profile\_id IS NOT NULL;

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

CREATE INDEX idx\_media\_asset\_type\_status ON public.media (media\_asset\_type, media\_status);

CREATE INDEX idx\_media\_tags ON public.media USING GIN (tags);

CREATE INDEX idx\_media\_deleted\_at ON public.media (deleted\_at) WHERE deleted\_at IS NOT NULL;

CREATE INDEX idx\_media\_last\_linked\_or\_used\_at ON public.media (last\_linked\_or\_used\_at) WHERE last\_linked\_or\_used\_at IS NOT NULL;

-- Comments

COMMENT ON TABLE public.media IS 'Central metadata repository for media files, referencing files in Supabase Storage and including audit trails. Version 2.2.';

COMMENT ON COLUMN public.media.uploader\_profile\_id IS 'Profile ID of the user who uploaded this media (serves as created\_by). If uploader''s profile is deleted, this becomes NULL.';

COMMENT ON COLUMN public.media.image\_variants\_json IS 'JSONB storing relative paths/identifiers for optimized versions (e.g., thumbnails, webp). Populated by the application layer/backend. Example: {"thumb\_s": "variants/uuid\_thumb\_s.webp", "display\_l": "variants/uuid\_display\_l.jpg"}.';

COMMENT ON COLUMN public.media.default\_alt\_text IS 'Default alternative text in the primary reference language for accessibility (WCAG). (Translatable via ''translations'' table).';

COMMENT ON COLUMN public.media.default\_caption IS 'Default caption in the primary reference language for display. (Translatable via ''translations'' table).';

COMMENT ON COLUMN public.media.updated\_by\_profile\_id IS 'Profile ID of the user who last significantly updated this media record (e.g., status change, key metadata update). ON DELETE SET NULL.';

COMMENT ON COLUMN public.media.last\_linked\_or\_used\_at IS 'Timestamp when this media item was last actively linked. Updated by triggers on linking tables. Useful for identifying orphaned media.';

4. Triggers/Functions

-- Trigger for general updated\_at timestamp

CREATE TRIGGER handle\_media\_updated\_at

BEFORE UPDATE ON public.media

FOR EACH ROW

EXECUTE FUNCTION extensions.moddatetime('updated\_at'); -- Standard Supabase trigger for updated\_at

-- Trigger function to clean up related translations on delete

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

RETURNS TRIGGER AS $$

BEGIN

DELETE FROM public.translations

WHERE table\_identifier = 'media'

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

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

-- Apply the cleanup trigger for translations

CREATE TRIGGER trigger\_cleanup\_media\_translations

AFTER DELETE ON public.media

FOR EACH ROW

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

-- Conceptual Trigger function to update 'last\_linked\_or\_used\_at' (to be called by triggers on media \*linking\* tables)

-- (Definition as provided previously)

/\*

CREATE OR REPLACE FUNCTION public.update\_media\_last\_linked\_timestamp() RETURNS TRIGGER AS $$ ... $$ LANGUAGE plpgsql SECURITY DEFINER;

\*/

5. JSON Schema Mirror

{

"title": "media\_item",

"description": "Master record for a media asset, including metadata, references to cloud storage, and audit trails. Content in 'default\_alt\_text' and 'default\_caption' is in the primary reference language. Version 2.2.",

"type": "object",

"properties": {

"id": { "type": "string", "format": "uuid", "description": "PK. Unique identifier.", "default": "gen\_random\_uuid()", "readOnly": true },

"uploader\_profile\_id": { "type": ["string", "null"], "format": "uuid", "description": "FK to profiles.id of uploader (created\_by)." },

"storage\_bucket\_name": { "type": "string", "description": "Supabase Storage bucket name." },

"storage\_object\_path\_original": { "type": "string", "description": "Unique path to original object in bucket." },

"file\_name\_original": { "type": "string", "description": "Original uploader filename." },

"file\_mime\_type": { "type": "string", "description": "MIME type of original file." },

"file\_size\_bytes\_original": { "type": ["integer", "null"], "format": "int64", "description": "Size of original file in bytes.", "minimum": 1 },

"image\_width\_px\_original": { "type": ["integer", "null"], "description": "Original width in pixels (for images).", "minimum": 1 },

"image\_height\_px\_original": { "type": ["integer", "null"], "description": "Original height in pixels (for images).", "minimum": 1 },

"image\_variants\_json": {

"type": ["object", "null"],

"description": "JSONB with paths to optimized image versions. Populated by app/backend. Ex: {\"thumb\_s\": \"path/to\_thumb.webp\"}.",

"examples": [ { "thumb\_s\_webp": "variants/uuid\_thumb\_s.webp", "display\_l\_jpeg": "variants/uuid\_display\_l.jpg" } ]

},

"default\_alt\_text": { "type": ["string", "null"], "description": "Default alt text (primary lang). Translatable." },

"default\_caption": { "type": ["string", "null"], "description": "Default caption (primary lang). Translatable." },

"attribution\_text": { "type": ["string", "null"], "description": "Credit/source. Potentially translatable." },

"attribution\_url": { "type": ["string", "null"], "format": "url", "pattern": "^https?://.+", "description": "URL to source/license." },

"licence": { "$ref": "#/definitions/media\_licence\_enum", "default": "all\_rights\_reserved" },

"media\_asset\_type": { "$ref": "#/definitions/media\_asset\_type\_enum", "default": "image" },

"media\_status": { "$ref": "#/definitions/media\_status\_enum", "default": "pending\_review" },

"tags": { "type": ["array", "null"], "items": { "type": "string" }, "description": "Keywords/tags for categorization." },

"checksum\_sha256\_original": { "type": ["string", "null"], "description": "SHA256 checksum of original file for de-duplication. Unique if not NULL." },

"dominant\_color\_hex": { "type": ["string", "null"], "pattern": "^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$", "description": "Optional dominant color of image (hex)." },

"last\_linked\_or\_used\_at": { "type": ["string", "null"], "format": "date-time", "description": "Timestamp last actively linked. For orphaned media cleanup." },

"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 metadata update.", "readOnly": true },

"updated\_by\_profile\_id": { "type": ["string", "null"], "format": "uuid", "description": "FK to profiles.id of last significant updater." },

"deleted\_at": { "type": ["string", "null"], "format": "date-time", "description": "Timestamp for soft deletion.", "readOnly": true }

},

"required": [ "id", "storage\_bucket\_name", "storage\_object\_path\_original", "file\_name\_original", "file\_mime\_type", "media\_asset\_type", "media\_status" ],

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

"unique\_constraints": [

{"columns": ["storage\_object\_path\_original"], "name": "uq\_media\_storage\_object\_path"},

{"columns": ["checksum\_sha256\_original"], "name": "uq\_media\_checksum\_original", "allow\_null\_results": true}

],

"foreign\_keys": [

{"columns": ["uploader\_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"}

],

"definitions": {

"media\_asset\_type\_enum": { "type": "string", "enum": ["image", "document\_pdf", "audio\_clip", "video\_clip", "gpx\_file", "other\_file"] },

"media\_licence\_enum": { "type": "string", "enum": [ "all\_rights\_reserved", "cc\_by", "cc\_by\_sa", "cc\_by\_nd", "cc\_by\_nc", "cc\_by\_nc\_sa", "cc\_by\_nc\_nd", "cc0\_public\_domain", "uploader\_owns\_contact\_for\_use", "official\_permission\_granted", "unknown\_licence" ]},

"media\_status\_enum": { "type": "string", "enum": [ "processing\_upload", "pending\_review", "published\_approved", "rejected\_hidden", "archived", "error\_uploading" ]}

}

}

6. Relationships & Integrity

Primary Key: `id` (uuid).

Foreign Keys:

• `uploader\_profile\_id` references `public.profiles(id)`: ON DELETE SET NULL (serves as `created\_by`).

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

Linking: Central repository, referenced by other linking tables (e.g., `waypoint\_media`) or direct FKs.

Orphaned Media Management: `last\_linked\_or\_used\_at` is updated by triggers on linking tables.

7. Multilingual Strategy

Translatable Fields: `default\_alt\_text`, `default\_caption`, optionally `attribution\_text` via `public.translations`. `AFTER DELETE` trigger on `media` cleans related translations.

8. Role-Based Workflow & RLS Notes

Key Fields: `uploader\_profile\_id`, `media\_status`, `updated\_by\_profile\_id`, `deleted\_at`.

RLS Policies: (As defined in Media Spec v2.1 - public read published, authenticated insert, user manage own, moderator/admin update, admin delete). Population of `updated\_by\_profile\_id` should be handled by application logic during moderation actions covered by RLS.

9. ENUM vs Lookup Discussion

🟢 ENUMs (`media\_asset\_type\_enum`, `media\_licence\_enum`, `media\_status\_enum`) are appropriate.

10. UI/UX Enablement

Columns for UI: `image\_variants\_json`, `default\_alt\_text` (translated), `default\_caption` (translated), `dominant\_color\_hex`, `media\_status`, `tags`.

Performance: Indexes on `tags` (GIN), `(media\_asset\_type, media\_status)`, FKs, `last\_linked\_or\_used\_at`.

11. Auditing & Lifecycle Management

Audit Columns: `uploader\_profile\_id` (as created\\_by), `created\_at`, `updated\_at`, `updated\_by\_profile\_id`.

Soft Deletes: `deleted\_at`. File storage cleanup is a separate background process.

12. Scalability & Future-Proofing

Scalability: UUID PKs, Supabase Storage, JSONB are scalable. Partitioning V2+ option.

Future-Proofing: ENUMs, `tags`, `jsonb` offer flexibility. `last\_linked\_or\_used\_at` aids maintenance.

13. Seed Data

⚪ N/A - This table stores transactional media metadata, not predefined master data requiring seeding.

14. Next-Action Checklist

• 🔴 Implement ENUM Types.

• 🔴 Create Table: Execute DDL for `public.media`.

• 🔴 Create Indexes.

• 🔴 Implement `updated\_at` Trigger (e.g., `extensions.moddatetime`).

• 🔴 Implement Translations Cleanup Trigger: Create and apply `trigger\_cleanup\_media\_translations` (AFTER `public.translations` table exists).

• 🟠 Develop Backend Media Processing Logic: For uploads, checksums, optimization, populating `image\_variants\_json`, and metadata extraction. This includes defining the structure and keys for `image\_variants\_json`.

• 🟠 Define Population Logic for `updated\_by\_profile\_id`: Ensure application logic or specific triggers populate this during moderation/admin updates.

• 🟠 Plan Orphaned Media File Cleanup Process from Storage.

• 🟠 Implement Triggers for `last\_linked\_or\_used\_at`: On all media \*linking\* tables.

• 🟢 Consider Full-Text Search on text fields (`default\_alt\_text`, `default\_caption`, `tags`, `file\_name\_original`) for V2+.