## [**https://gemini.google.com/u/1/app/980a7af1dc7f079a**](https://gemini.google.com/u/1/app/980a7af1dc7f079a)

https://gemini.google.com/u/1/app/117b6ecbb3fe7734

\* \* \* \* \*

Updated Production-Ready Specification: Content Statuses Master Table (Version 1.2)

-----------------------------------------------------------------------------------

This document details the structure, purpose, and considerations for the `content\_statuses\_master` table. This table is fundamental for managing the lifecycle and visibility of content across the platform. Version 1.2 updates RLS policies for consistency with the platform-wide security architecture. Version 1.1 incorporated the `is\_active` flag for the lifecycle management of status definitions themselves and included standard V2 enhancements.

1\. Purpose & Primary Use-Cases

The `content\_statuses\_master` table defines the various states content can be in (e.g., "Draft," "Published"). Its purpose is to standardize content visibility and workflow progression, allowing separation of work-in-progress from public content and supporting moderation. Key uses include admin content lifecycle management and system filtering for public display.

2\. Updated Schema (Markdown Table)

| column | data\_type | constraints | description |

| id | integer | Primary Key (Generated as identity always) | Unique identifier for the content status. |

| 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., 'draft', 'published'). Snake\_case, max 50 chars. |

| label | text | Not Null, CHECK (length(label) > 0 AND length(label) &lt;= 100) | Human-readable name for the status. (Translatable via `public.translations`). Primary reference language (English). Max 100 chars. |

| description | text | Nullable | Optional description of the status. (Translatable via `public.translations`). Primary reference language (English). |

| is\_publicly\_visible | boolean | Not Null, Default false | Indicates if \*content with this status\* is generally visible to non-authenticated/public users (e.g., 'published' would be true). |

| sort\_order | integer | Not Null, Default 0 | Determines the display order of statuses in UI selection lists or workflow diagrams. Lower numbers appear first. |

| is\_active | boolean | Not Null, Default true | True if the status definition itself is active and available for assignment to content; false if this status definition is retired. |

| created\_at | timestamp with time zone | Not Null, Default now() | Timestamp of record creation. |

| updated\_at | timestamp with time zone | Not Null, Default now() | Timestamp of last update (auto-updated by trigger). |

| created\_by\_profile\_id | uuid | Nullable, Foreign Key to profiles(id) ON DELETE SET NULL | Profile ID of the user who created the record (typically an admin for initial setup). |

| updated\_by\_profile\_id | uuid | Nullable, Foreign Key to profiles(id) ON DELETE SET NULL | Profile ID of the user who last updated the record. |

3\. PostgreSQL DDL

SQL

```

-- Assumes public.profiles table exists and has a `roles TEXT[]` column.

-- Assumes RLS helper function public.has\_role(TEXT) RETURNS BOOLEAN exists.

-- Assumes public.set\_current\_timestamp\_updated\_at() function exists.

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

CREATE TABLE public.content\_statuses\_master (

id SERIAL 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,

is\_publicly\_visible 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.content\_statuses\_master IS 'Master list of content publication statuses (e.g., draft, published, archived). `label` and `description` are translatable. Version 1.2';

COMMENT ON COLUMN public.content\_statuses\_master.id IS 'Unique identifier for the content status.';

COMMENT ON COLUMN public.content\_statuses\_master.code IS 'Short, stable, machine-readable code (snake\_case). Max 50 chars. E.g., ''draft'', ''published''.';

COMMENT ON COLUMN public.content\_statuses\_master.label IS 'Human-readable English label for UI and translation base. Max 100 chars. (Translatable via public.translations).';

COMMENT ON COLUMN public.content\_statuses\_master.description IS 'Optional English description of the status. (Translatable via public.translations).';

COMMENT ON COLUMN public.content\_statuses\_master.is\_publicly\_visible IS 'Boolean flag indicating if content \*assigned\* this status is generally visible to the public.';

COMMENT ON COLUMN public.content\_statuses\_master.sort\_order IS 'Determines the display order in UI lists or workflow representations.';

COMMENT ON COLUMN public.content\_statuses\_master.is\_active IS 'True if the status definition itself is active and available for assignment to content; false if this status definition is retired. Defaults to true.';

COMMENT ON COLUMN public.content\_statuses\_master.created\_at IS 'Timestamp of record creation.';

COMMENT ON COLUMN public.content\_statuses\_master.updated\_at IS 'Timestamp of last update (auto-updated by trigger).';

COMMENT ON COLUMN public.content\_statuses\_master.created\_by\_profile\_id IS 'Profile ID of the user who created the record.';

COMMENT ON COLUMN public.content\_statuses\_master.updated\_by\_profile\_id IS 'Profile ID of the user who last updated the record.';

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_cs\_master\_created\_by ON public.content\_statuses\_master(created\_by\_profile\_id);

CREATE INDEX IF NOT EXISTS idx\_cs\_master\_updated\_by ON public.content\_statuses\_master(updated\_by\_profile\_id);

CREATE INDEX IF NOT EXISTS idx\_cs\_master\_is\_publicly\_visible ON public.content\_statuses\_master(is\_publicly\_visible);

CREATE INDEX IF NOT EXISTS idx\_cs\_master\_sort\_order ON public.content\_statuses\_master(sort\_order);

CREATE INDEX IF NOT EXISTS idx\_cs\_master\_is\_active ON public.content\_statuses\_master(is\_active);

-- Trigger for updated\_at

CREATE TRIGGER trigger\_content\_statuses\_master\_set\_updated\_at

BEFORE UPDATE ON public.content\_statuses\_master

FOR EACH ROW

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

COMMENT ON TRIGGER trigger\_content\_statuses\_master\_set\_updated\_at ON public.content\_statuses\_master IS 'Trigger to automatically update updated\_at timestamp on row modification.';

-- Trigger for orphan translation cleanup

CREATE TRIGGER trigger\_cleanup\_content\_statuses\_master\_translations

AFTER DELETE ON public.content\_statuses\_master

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_related\_translations('content\_statuses\_master');

COMMENT ON TRIGGER trigger\_cleanup\_content\_statuses\_master\_translations ON public.content\_statuses\_master IS 'Cleans up orphaned translations when a content status is deleted. The trigger function should pass OLD.id::TEXT as the row foreign key.';

```

4\. JSON Schema Mirror

JSON

```

{

"title": "content\_status\_master",

"description": "Master list of content publication statuses. `label` and `description` are translatable. Version 1.2",

"type": "object",

"properties": {

"id": {

"type": "integer",

"description": "Unique identifier for the content status. Primary Key.",

"readOnly": true

},

"code": {

"type": "string",

"description": "Short, stable, machine-readable code (snake\_case). Max 50 chars.",

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

"maxLength": 50

},

"label": {

"type": "string",

"description": "Human-readable English label for UI and translation base. Max 100 chars. (Translatable via public.translations).",

"maxLength": 100

},

"description": {

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

"description": "Optional English description of the status. (Translatable via public.translations)."

},

"is\_publicly\_visible": {

"type": "boolean",

"default": false,

"description": "Indicates if content \*assigned\* this status is generally visible to the public."

},

"sort\_order": {

"type": "integer",

"default": 0,

"description": "Determines display order in UI lists or workflow representations."

},

"is\_active": {

"type": "boolean",

"default": true,

"description": "True if the status definition itself is active and available for assignment to content; false if this status definition is 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.",

"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\_publicly\_visible",

"sort\_order",

"is\_active"

]

}

```

5\. Relationships & Integrity

- Primary Key: `id` (SERIAL -> INTEGER)

- Unique Constraint: `code` must be unique.

- Foreign Key References FROM other tables:

- `waypoints.content\_visibility\_status\_id` REFERENCES `public.content\_statuses\_master(id)` (ON DELETE RESTRICT).

- Other content tables (e.g., `articles`, `curated\_itineraries`) will also use this table for status management.

- 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.

- Data Integrity Notes:

- Deletion of a status definition will be restricted if any content item is actively using it. Retiring a status definition should be done by setting `is\_active = false`.

6\. Multilingual Strategy

- Translatable Fields:

- `label`: The English name for the status. (Translatable via `public.translations`)

- `description`: The English description for the status. (Translatable via `public.translations`)

- Non-Translatable Fields:

- `code` is a stable identifier and is not translated.

- `is\_publicly\_visible` and `is\_active` are system flags.

- Translation Management:

- An `AFTER DELETE` trigger (`trigger\_cleanup\_content\_statuses\_master\_translations`) is implemented to call `public.cleanup\_related\_translations` to remove orphaned translations.

7\. Role-Based Workflow & RLS Notes

- Content Management: This table defines the core lifecycle statuses for content. Definitions are managed by users with the `platform\_admin` role. Other authenticated users (like `admin`, `regional\_content\_manager`) need to be able to read these statuses to assign them to content items they manage.

- Lifecycle: Status definitions are retired by setting `is\_active = false`.

- RLS Policies:

- Enable RLS:

SQL

```

ALTER TABLE public.content\_statuses\_master ENABLE ROW LEVEL SECURITY;

```

- Anonymous & Authenticated Users (Read-Only for Active Status Definitions): All users who might need to understand or assign statuses (including anonymous users if UI elements display status labels from here for public content) need to read active status definitions.

SQL

```

CREATE POLICY "Allow public read access to active content status definitions"

ON public.content\_statuses\_master FOR SELECT

USING (is\_active = true);

```

- Platform Administrators (Full Control over Status Definitions):

SQL

```

CREATE POLICY "Allow platform\_admins full access to content status definitions"

ON public.content\_statuses\_master FOR ALL

USING (public.has\_role('platform\_admin'))

WITH CHECK (public.has\_role('platform\_admin'));

```

- Note: Ensure the `public.has\_role(TEXT)` function is defined and correctly checks the `roles` array in the `public.profiles` table for the current `auth.uid()`. This RLS implies that only `platform\_admin` can create, update, or delete these master status definitions. Other roles select from the active ones.

8\. ENUM vs Lookup Discussion

- 🟢 Decision: This `content\_statuses\_master` table correctly promotes an original ENUM concept for content status management.

- Reasoning: A master lookup table provides clarity, richness (additional attributes like `description`, `is\_publicly\_visible`, `is\_active`, `sort\_order`), i18n, maintainability, workflow support, and auditability.

9\. UI/UX Enablement

- `label` (translated): Used for display names in status selectors in content management forms, and potentially in workflow diagrams or status indicators.

- `is\_publicly\_visible`: Crucial for system logic to determine what content RLS policies should expose to the public.

- `sort\_order`: Ensures a logical presentation of statuses in dropdowns or workflow step displays.

- `is\_active`: UI for content management should primarily offer active status definitions for assignment.

- Default Value: Tables like `waypoints` correctly default their status FK to the 'draft' status ID from this table.

10\. Key Considerations & Definitions

- Core Statuses: The initial set ('draft', 'pending\_review', 'published', 'archived', 'private') covers common content lifecycles.

- `is\_publicly\_visible` Flag: This is a key field that drives how RLS policies should be constructed for public-facing content.

- `is\_active` Flag: Key for managing the lifecycle of the status definitions themselves.

- Workflow Implications: Changes to statuses can impact content management workflows.

11\. Scalability & Future-Proofing

- Lookup Table Structure: Easily supports adding new content statuses or modifying attributes of existing ones.

- Manageable List: The number of distinct content statuses is expected to be small.

- Audit Fields & `is\_active` flag: Provide change history and robust lifecycle management.

12\. Next-Action Checklist

- 🔴 Define Global RLS Helper Functions: Ensure `public.has\_role(TEXT)` is implemented securely and available.

- 🔴 Create Table: Execute the DDL for `public.content\_statuses\_master` (Version 1.2).

- 🔴 Apply Triggers: Ensure `set\_current\_timestamp\_updated\_at` and `cleanup\_related\_translations` triggers are created and applied. Confirm `cleanup\_related\_translations` function is correctly set up.

- 🔴 Initial Population: Insert defined content statuses (e.g., draft, published). Ensure `is\_active = true`, `is\_publicly\_visible` is set appropriately, and `created\_by\_profile\_id` is set (e.g., to a system admin ID). Crucially, ensure the 'draft' status ID exists before tables like `waypoints` are created due to its default FK constraint.

- 🟠 RLS Policies: Implement and thoroughly test the RLS policies defined in section 7.

- 🟢 Translation Entries: Prepare initial English entries for `label` and `description` in `public.translations`.

\* \* \* \* \*