https://gemini.google.com/u/1/app/951c86a3d09aaadb

\* \* \* \* \*

Production-Ready Specification: public.public\_active\_segment\_warnings\_view

Version: 1.0

Date: May 18, 2025

1. Purpose & Primary Use-Cases

- Mission Statement: To provide a simplified, secure, and consolidated read-only interface for accessing currently active and publicly published segment warnings. This view joins essential details from `segment\_warnings` with related master tables (`warning\_types\_master`, `warning\_severities\_master`, `warning\_source\_types\_master`) to present user-friendly information.

- Key User-Story Touchpoints:

- Pilgrim (Anna): Applications (web/mobile) used by pilgrims will query this view to display warnings relevant to their planned or current journey, including type, severity, description, and visual cues like icons and colors.

- Public API: Endpoints that expose segment warning data to third-party services or public frontends would benefit from querying this pre-joined and pre-filtered view.

- Developers: Simplifies frontend development by providing a ready-to-use data structure for warnings, reducing the need for complex client-side joins or multiple API calls.

2. Schema (Exposed Columns by the View)

| Column | Data Type | Source Table(s) | Description |

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

| warning\_id | bigint | segment\_warnings.id | Unique identifier of the segment warning. |

| segment\_id | bigint | segment\_warnings.segment\_id | Identifier of the trail segment the warning applies to. |

| title | text | segment\_warnings.title | Headline of the warning (primary reference language). (Translatable via public.translations). |

| description\_message | text | segment\_warnings.description\_message | Detailed description of the warning (primary reference language). (Translatable via public.translations). |

| location\_on\_segment\_description| text | segment\_warnings.location\_on\_segment\_description| Textual description of the location on the segment (primary reference language). (Translatable). |

| location\_on\_segment\_km\_approx | real | segment\_warnings.location\_on\_segment\_km\_approx | Approximate kilometer marker of the warning on the segment. |

| location\_on\_segment\_geom | geometry(PointZ,4326) | segment\_warnings.location\_on\_segment\_geom | PostGIS geometry pinpointing the warning's location. |

| detour\_information\_url | text | segment\_warnings.detour\_information\_url | URL for detour information, if available. |

| detour\_description\_notes | text | segment\_warnings.detour\_description\_notes | Notes about the detour (primary reference language). (Translatable via public.translations). |

| date\_warning\_reported | timestamptz | segment\_warnings.date\_warning\_reported | Timestamp when the warning was reported. |

| date\_warning\_effective\_from | timestamptz | segment\_warnings.date\_warning\_effective\_from | Timestamp from when the warning is considered effective. |

| date\_warning\_expected\_resolution| timestamptz | segment\_warnings.date\_warning\_expected\_resolution| Timestamp when the warning is expected to be resolved. |

| warning\_type\_code | text | warning\_types\_master.code | Machine-readable code for the warning type. |

| warning\_type\_name | text | warning\_types\_master.display\_name | Display name of the warning type (primary reference language). (Translatable). |

| warning\_type\_icon\_identifier | text | warning\_types\_master.icon\_identifier | UI icon identifier for the warning type. |

| warning\_severity\_code | text | warning\_severities\_master.code | Machine-readable code for the warning severity. |

| warning\_severity\_name | text | warning\_severities\_master.display\_name | Display name of the warning severity (primary reference language). (Translatable). |

| warning\_severity\_ui\_color\_hex | text | warning\_severities\_master.ui\_color\_hex | UI color hex code for the warning severity. |

| warning\_source\_type\_name | text | warning\_source\_types\_master.display\_name | Display name of the warning source type (primary reference language, if included). (Translatable). |

| primary\_image\_media\_id | uuid | segment\_warnings.primary\_image\_media\_id | UUID of the primary image associated with the warning, linking to public.media. |

3. PostgreSQL DDL

SQL

```

CREATE OR REPLACE VIEW public.public\_active\_segment\_warnings\_view AS

SELECT

sw.id AS warning\_id,

sw.segment\_id,

sw.title, -- Primary reference language text

sw.description\_message, -- Primary reference language text

sw.location\_on\_segment\_description, -- Primary reference language text

sw.location\_on\_segment\_km\_approx,

sw.location\_on\_segment\_geom,

sw.detour\_information\_url,

sw.detour\_description\_notes, -- Primary reference language text

sw.date\_warning\_reported,

sw.date\_warning\_effective\_from,

sw.date\_warning\_expected\_resolution,

wtm.code AS warning\_type\_code,

wtm.display\_name AS warning\_type\_name, -- Primary reference language text

wtm.icon\_identifier AS warning\_type\_icon\_identifier,

wsm.code AS warning\_severity\_code,

wsm.display\_name AS warning\_severity\_name, -- Primary reference language text

wsm.ui\_color\_hex AS warning\_severity\_ui\_color\_hex,

wstm.display\_name AS warning\_source\_type\_name, -- Primary reference language text

sw.primary\_image\_media\_id

-- To include direct image paths, an additional JOIN to public.media would be needed:

-- m.storage\_object\_path\_original AS primary\_image\_original\_url,

-- m.image\_variants\_json ->> 'thumb\_s\_webp' AS primary\_image\_thumb\_s\_webp\_url

FROM

public.segment\_warnings sw

JOIN

public.warning\_types\_master wtm ON sw.warning\_type\_id = wtm.id

JOIN

public.warning\_severities\_master wsm ON sw.warning\_severity\_id = wsm.id

LEFT JOIN -- Assuming warning\_source\_type\_id can be NULL

public.warning\_source\_types\_master wstm ON sw.warning\_source\_type\_id = wstm.id

-- LEFT JOIN public.media m ON sw.primary\_image\_media\_id = m.id -- Example join for media paths

WHERE

sw.is\_currently\_active = true

AND sw.workflow\_status\_code = 'published' -- Or the appropriate code for "publicly visible"

AND wtm.is\_active = true

AND wsm.is\_active = true

AND (wstm.id IS NULL OR wstm.is\_active = true); -- Only include if source type exists and is active

COMMENT ON VIEW public.public\_active\_segment\_warnings\_view IS 'Provides a consolidated, filtered view of currently active and published segment warnings with essential details from related master tables. Intended for public read access. Version 1.0.';

```

4. JSON Schema Mirror (Output Structure Description)

JSON

```

{

"title": "public\_active\_segment\_warning",

"description": "Structure of a warning object as exposed by the public\_active\_segment\_warnings\_view. All text fields are in the primary reference language unless otherwise specified by application-level translation.",

"type": "object",

"properties": {

"warning\_id": { "type": "integer", "format": "int64", "description": "Unique ID of the warning." },

"segment\_id": { "type": "integer", "format": "int64", "description": "ID of the affected segment." },

"title": { "type": "string", "description": "Warning title (primary reference language)." },

"description\_message": { "type": "string", "description": "Detailed warning description (primary reference language)." },

"location\_on\_segment\_description": { "type": ["string", "null"], "description": "Textual location on segment (primary reference language)." },

"location\_on\_segment\_km\_approx": { "type": ["number", "null"], "description": "Approximate KM marker." },

"location\_on\_segment\_geom": { "type": ["object", "null"], "description": "GeoJSON PointZ geometry." },

"detour\_information\_url": { "type": ["string", "null"], "format": "url", "description": "URL for detour info." },

"detour\_description\_notes": { "type": ["string", "null"], "description": "Detour notes (primary reference language)." },

"date\_warning\_reported": { "type": "string", "format": "date-time", "description": "Date warning was reported." },

"date\_warning\_effective\_from": { "type": ["string", "null"], "format": "date-time", "description": "Date warning is effective from." },

"date\_warning\_expected\_resolution": { "type": ["string", "null"], "format": "date-time", "description": "Date warning is expected to be resolved." },

"warning\_type\_code": { "type": "string", "description": "Code of the warning type." },

"warning\_type\_name": { "type": "string", "description": "Name of the warning type (primary reference language)." },

"warning\_type\_icon\_identifier": { "type": ["string", "null"], "description": "Icon for the warning type." },

"warning\_severity\_code": { "type": "string", "description": "Code of the warning severity." },

"warning\_severity\_name": { "type": "string", "description": "Name of the warning severity (primary reference language)." },

"warning\_severity\_ui\_color\_hex": { "type": ["string", "null"], "description": "UI color for the warning severity." },

"warning\_source\_type\_name": { "type": ["string", "null"], "description": "Name of the warning source type (primary reference language)." },

"primary\_image\_media\_id": { "type": ["string", "null"], "format": "uuid", "description": "UUID of the primary image in the media table." }

}

}

```

5. Relationships & Integrity

- This view directly depends on the following tables: `public.segment\_warnings`, `public.warning\_types\_master`, `public.warning\_severities\_master`, and `public.warning\_source\_types\_master`.

- It may optionally join `public.media` if image path details are included directly in the view.

- The integrity of the data presented by the view relies entirely on the integrity constraints (Primary Keys, Foreign Keys, CHECK constraints) defined on these underlying base tables.

- The view itself does not store data and thus has no independent integrity constraints.

6. Multilingual Strategy

- The view exposes textual fields (e.g., `title`, `description\_message`, `warning\_type\_name`, `warning\_severity\_name`) in the platform's primary reference language (English).

- Actual translation into other languages for end-users must be handled by the application layer. The application would query the `public.translations` table using the `warning\_id` (for `segment\_warnings` fields) or the respective master table ID/code (for `warning\_types\_master.display\_name`, etc.) and the target `language\_code` to fetch translated strings.

7. Role-Based Workflow & RLS Notes

- Purpose: This view is intended for public read access.

- Filtering: The `WHERE` clause in the view definition (`sw.is\_currently\_active = true AND sw.workflow\_status\_code = 'published' AND master\_table.is\_active = true`) is the primary mechanism for ensuring only appropriate data is exposed.

- RLS Policy on View:

SQL

```

CREATE POLICY "public\_select\_on\_active\_segment\_warnings\_view"

ON public.public\_active\_segment\_warnings\_view

FOR SELECT

TO anonymous, authenticated -- Granting to general roles

USING (true);

```

- RLS on underlying tables will still be evaluated by PostgreSQL when a user queries the view, providing an additional layer of security. However, since the view's `WHERE` clause is already restrictive for public data, the RLS on base tables for a public user querying this view should ideally align or be less restrictive for these specific "published" records.

8. ENUM vs Lookup Discussion

- ⚪ N/A (This is a view, not a table using ENUMs or lookups directly for its own definition). It consumes data from tables that correctly use lookup tables instead of ENUMs.

9. UI/UX Enablement

- Provides a denormalized, ready-to-consume structure for displaying segment warnings in user interfaces (maps, lists, detail pages).

- Simplifies frontend logic by pre-joining necessary information (type name, severity name, colors, icons).

- The filtering for active and published warnings ensures UIs only show relevant, current information to pilgrims by default.

10. Key Considerations & Definitions

- `published` status: The view assumes `'published'` is the specific `workflow\_status\_code` representing publicly visible warnings. This code must exist in `public.workflow\_statuses\_master`.

- `is\_active` flags: The view correctly filters on `is\_active = true` for the joined master tables (`warning\_types\_master`, `warning\_severities\_master`, `warning\_source\_types\_master`) ensuring that definitions for types, severities, or sources that have been retired are not used for displaying current warnings.

- Performance: While views simplify queries, complex views with many joins can have performance implications. The joins in this view are on indexed foreign keys to relatively small master tables, which should generally perform well. For very high-traffic scenarios, monitoring and potential optimization (e.g., ensuring base table indexes are optimal, or considering a materialized view if stale data is acceptable for short periods) might be needed.

11. Auditing & Lifecycle Management

- ⚪ N/A for the view itself. Auditing and lifecycle are managed by the underlying base tables (`segment\_warnings`, etc.). This view only presents data based on the current state of those tables.

12. Scalability & Future-Proofing

- Scalability: The view's performance will scale with the performance of queries on its base tables, particularly `segment\_warnings`. As `segment\_warnings` grows, indexing and the filtering criteria are crucial.

- Future-Proofing: The view provides a level of abstraction. If underlying table structures change slightly but the conceptual data needed for display remains, the view definition can be updated, potentially minimizing impact on client applications. Adding new, commonly requested fields to the view is straightforward.

13. Seed Data

- ⚪ N/A (Views do not contain data themselves).

14. Next-Action Checklist

- 🔴 Finalize View Columns: Confirm if direct media paths (e.g., `primary\_image\_original\_url`, `primary\_image\_thumb\_s\_webp\_url`) by joining `public.media` are required in the view, or if `primary\_image\_media\_id` is sufficient for the application to fetch media details separately.

- 🔴 Implement DDL: Execute the `CREATE OR REPLACE VIEW` statement.

- 🔴 Test View: Query the view with various scenarios (no warnings, multiple warnings, warnings with/without optional fields like geometry or source type) to ensure correctness.

- 🔴 Implement RLS Policy: Apply the RLS policy to the view.

- 🟢 Performance Testing: Under expected load, test the performance of queries against this view.

- 🟢 Documentation: Document this view for API developers and frontend developers who will consume it.

\* \* \* \* \*