<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>

\* \* \* \* \*

### Updated Production-Ready Specification

Table Name: `public.trail\_regions`

1\. Purpose & Primary Use-Cases

This table links trails to the geographical regions they traverse, defining the sequence of regions for a trail. It allows for region-specific details such as significance notes, operational status updates for the trail within that region, and flags for entry/exit points, thus providing contextual geographical information essential for pilgrims and content managers.

Key User-Story Touchpoints:

- Pilgrim (Anna) - Story A2 (Daily Stage Understanding): Helps contextualize which region a particular part of the trail falls into.

- Pilgrim (Anna) - Story A5 (Checking Trail Conditions): `regional\_trail\_status` provides crucial, localized updates that can override the general trail status for a specific region.

- Regional Content Manager (Sofia) - Story C1 & C2: Allows Sofia to update `regional\_trail\_status` and `regional\_significance\_notes` for regions she manages, and forms part of the basis for determining her scope of responsibility.

- Platform Administrator (Admin Team) - Story D1: Admins manage these trail-to-region links and their attributes.

2\. Schema

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

| `id` | `bigint` | PRIMARY KEY, GENERATED ALWAYS AS IDENTITY | Surrogate unique identifier for this trail-region link record, simplifying linking from other tables (e.g., `translations`). |

| `trail\_id` | `bigint` | NOT NULL, REFERENCES `public.trails(id)` ON DELETE CASCADE | The ID of the trail this entry pertains to. |

| `region\_id` | `bigint` | NOT NULL, REFERENCES `public.regions(id)` ON DELETE CASCADE | The ID of the region this trail passes through. |

| `display\_order` | `integer` | NOT NULL, CHECK (`display\_order` > 0) | The sequential, 1-based order in which the trail passes through this region. |

| `regional\_significance\_notes` | `text` | | Describes the unique character or significance of this trail within this region (English base). (Translatable via `public.translations` using `trail\_regions.id`) |

| `is\_entry\_point\_region` | `boolean` | NOT NULL DEFAULT `false` | Flags if this region contains a common or official starting point for this trail. |

| `is\_exit\_point\_region` | `boolean` | NOT NULL DEFAULT `false` | Flags if this region contains a common or official ending point for this trail. |

| `is\_primary\_focus\_region` | `boolean` | NOT NULL DEFAULT `false` | Flags if this region is a central, most significant part of this trail's experience. |

| `estimated\_coverage\_percentage` | `real` | CHECK (`estimated\_coverage\_percentage` IS NULL OR (`estimated\_coverage\_percentage` >= 0 AND `estimated\_coverage\_percentage` &lt;= 100)) | Approximate percentage (0-100) of the trail's total length within this region. |

| `regional\_trail\_status` | `public.trail\_operational\_status\_enum` | | Specific operational status of the trail within this region; can override `trails.operational\_status`. |

| `regional\_status\_last\_checked` | `timestamptz` | | Timestamp when `regional\_trail\_status` or notes were last verified/updated. |

| `regional\_status\_updated\_by\_profile\_id` | `uuid` | REFERENCES `public.profiles(id)` ON DELETE SET NULL | Profile ID of user who last updated regional status/notes. |

| `created\_at` | `timestamptz` | NOT NULL DEFAULT `now()` | Timestamp of record creation. |

| `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 of last update (auto-updated by trigger). |

| `updated\_by\_profile\_id` | `uuid` | REFERENCES `public.profiles(id)` ON DELETE SET NULL | Profile ID of user who last made any update to this record. |

| | | UNIQUE (`trail\_id`, `region\_id`) | Ensures a trail-region pair is unique. |

| | | UNIQUE (`trail\_id`, `display\_order`) | Ensures `display\_order` is unique for each trail. |

3\. PostgreSQL DDL

SQL

```

-- Assumes 'public.trails' (BIGINT PK), 'public.regions' (BIGINT PK),

-- 'public.profiles' (UUID PK) tables exist.

-- Assumes 'public.trail\_operational\_status\_enum' is created.

CREATE TABLE public.trail\_regions (

id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY, -- [cite: 9, 10]

trail\_id BIGINT NOT NULL REFERENCES public.trails(id) ON DELETE CASCADE, -- [cite: 11]

region\_id BIGINT NOT NULL REFERENCES public.regions(id) ON DELETE CASCADE, -- [cite: 12]

display\_order INTEGER NOT NULL CHECK (display\_order > 0), -- [cite: 14]

regional\_significance\_notes TEXT, -- [cite: 15]

is\_entry\_point\_region BOOLEAN NOT NULL DEFAULT FALSE, -- [cite: 16]

is\_exit\_point\_region BOOLEAN NOT NULL DEFAULT FALSE, -- [cite: 17]

is\_primary\_focus\_region BOOLEAN NOT NULL DEFAULT FALSE, -- [cite: 19]

estimated\_coverage\_percentage REAL CHECK (estimated\_coverage\_percentage IS NULL OR (estimated\_coverage\_percentage >= 0 AND estimated\_coverage\_percentage <= 100)), -- [cite: 21, 22]

regional\_trail\_status public.trail\_operational\_status\_enum, -- [cite: 23]

regional\_status\_last\_checked TIMESTAMPTZ, -- [cite: 24]

regional\_status\_updated\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 25]

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(), -- [cite: 27]

created\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 28]

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(), -- [cite: 30]

updated\_by\_profile\_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL, -- [cite: 31]

UNIQUE (trail\_id, region\_id), -- [cite: 33]

UNIQUE (trail\_id, display\_order) -- [cite: 34]

);

-- Indexes

CREATE INDEX IF NOT EXISTS idx\_trail\_regions\_trail\_id ON public.trail\_regions(trail\_id); -- [cite: 39]

CREATE INDEX IF NOT EXISTS idx\_trail\_regions\_region\_id ON public.trail\_regions(region\_id); -- [cite: 39]

CREATE INDEX IF NOT EXISTS idx\_trail\_regions\_regional\_trail\_status ON public.trail\_regions(regional\_trail\_status); -- [cite: 40]

-- Trigger for audit columns

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

RETURNS TRIGGER AS $$

BEGIN

NEW.updated\_at = NOW(); -- [cite: 41]

NEW.updated\_by\_profile\_id = auth.uid(); -- General updater for any change to the row [cite: 41]

IF (TG\_OP = 'INSERT') THEN

NEW.created\_by\_profile\_id = auth.uid(); -- [cite: 42]

IF NEW.regional\_trail\_status IS NOT NULL OR NEW.regional\_significance\_notes IS NOT NULL THEN

NEW.regional\_status\_updated\_by\_profile\_id = auth.uid(); -- [cite: 42]

NEW.regional\_status\_last\_checked = NOW(); -- [cite: 43]

END IF;

ELSIF (TG\_OP = 'UPDATE') THEN

NEW.created\_at = OLD.created\_at; -- [cite: 44]

NEW.created\_by\_profile\_id = OLD.created\_by\_profile\_id; -- [cite: 45]

IF (NEW.regional\_trail\_status IS DISTINCT FROM OLD.regional\_trail\_status) OR

(NEW.regional\_significance\_notes IS DISTINCT FROM OLD.regional\_significance\_notes) THEN

NEW.regional\_status\_updated\_by\_profile\_id = auth.uid(); -- [cite: 46]

NEW.regional\_status\_last\_checked = NOW(); -- [cite: 46]

ELSIF NEW.regional\_status\_last\_checked IS DISTINCT FROM OLD.regional\_status\_last\_checked THEN

NEW.regional\_status\_updated\_by\_profile\_id = auth.uid(); -- [cite: 47]

ELSE

NEW.regional\_status\_updated\_by\_profile\_id = OLD.regional\_status\_updated\_by\_profile\_id; -- [cite: 48]

NEW.regional\_status\_last\_checked = OLD.regional\_status\_last\_checked; -- [cite: 48]

END IF;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER; -- [cite: 49]

CREATE TRIGGER trigger\_trail\_regions\_modification\_meta

BEFORE INSERT OR UPDATE ON public.trail\_regions

FOR EACH ROW

EXECUTE FUNCTION public.set\_trail\_region\_modification\_meta(); -- [cite: 50]

-- Comments

COMMENT ON TABLE public.trail\_regions IS 'Links trails to the geographical regions they traverse, defining sequence and region-specific trail attributes like status and significance notes. Version: V2.'; -- [cite: 50]

COMMENT ON COLUMN public.trail\_regions.id IS 'Surrogate unique identifier for the trail-region link. Version: V2.'; -- [cite: 51]

COMMENT ON COLUMN public.trail\_regions.trail\_id IS 'Foreign key to the public.trails table. Version: V2.'; -- [cite: 52]

COMMENT ON COLUMN public.trail\_regions.region\_id IS 'Foreign key to the public.regions table. Version: V2.'; -- [cite: 53]

COMMENT ON COLUMN public.trail\_regions.display\_order IS 'The sequential, 1-based order in which the trail passes through this region. Defines the regional progression of the trail. Version: V2.'; -- [cite: 54, 55]

COMMENT ON COLUMN public.trail\_regions.regional\_significance\_notes IS 'Describes the unique character, cultural significance, or specific pilgrim advice for this trail within this particular region (English base, translatable using trail\_regions.id). Version: V2.'; -- [cite: 15, 56]

COMMENT ON COLUMN public.trail\_regions.regional\_trail\_status IS 'Specific operational status of the trail section within this region. Can override the general trails.operational\_status if more granular information is available. Version: V2.'; -- [cite: 23, 57]

COMMENT ON COLUMN public.trail\_regions.regional\_status\_last\_checked IS 'Timestamp indicating when the regional\_trail\_status or regional\_significance\_notes were last actively verified or updated by a content manager. Version: V2.'; -- [cite: 24, 58]

COMMENT ON COLUMN public.trail\_regions.regional\_status\_updated\_by\_profile\_id IS 'Profile ID of the user who last made a substantive update to regional\_trail\_status or regional\_significance\_notes for this entry. Version: V2.'; -- [cite: 25, 59]

COMMENT ON COLUMN public.trail\_regions.created\_at IS 'Timestamp of when this trail-region link record was initially created. Version: V2.';

COMMENT ON COLUMN public.trail\_regions.created\_by\_profile\_id IS 'Profile ID of the user who initially created this trail-region link. Version: V2.';

COMMENT ON COLUMN public.trail\_regions.updated\_at IS 'Timestamp indicating when this record was last updated (auto-updated by trigger for any change). Version: V2.';

COMMENT ON COLUMN public.trail\_regions.updated\_by\_profile\_id IS 'Profile ID of the user who performed the most recent update on any field in this record. Version: V2.'; -- [cite: 31, 60]

```

4\. JSON Schema Mirror

JSON

```

{

"title": "trail\_region\_link",

"description": "Defines the many-to-many relationship between a trail and a region it traverses, including specific contextual information and status for that intersection. Uses a surrogate PK for easier linking. Version: V2.",

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64",

"description": "Surrogate unique identifier for this trail-region link record. Read-only.", -- [cite: 9, 10]

"readOnly": true

},

"trail\_id": {

"type": "integer",

"format": "int64",

"description": "Foreign key referencing the ID of the trail (public.trails.id)." -- [cite: 11, 62, 63]

},

"region\_id": {

"type": "integer",

"format": "int64",

"description": "Foreign key referencing the ID of the region (public.regions.id)." -- [cite: 12, 64]

},

"display\_order": {

"type": "integer",

"minimum": 1,

"description": "The sequential, 1-based order in which the trail passes through this region." -- [cite: 14, 65]

},

"regional\_significance\_notes": {

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

"description": "Notes on the trail's character or significance within this specific region (English base, translatable using trail\_regions.id)." -- [cite: 15, 66]

},

"is\_entry\_point\_region": {

"type": "boolean",

"default": false,

"description": "Flags if this region contains a common or official starting point for the trail." -- [cite: 16, 67]

},

"is\_exit\_point\_region": {

"type": "boolean",

"default": false,

"description": "Flags if this region contains a common or official ending point for the trail." -- [cite: 17, 68]

},

"is\_primary\_focus\_region": {

"type": "boolean",

"default": false,

"description": "Flags if this region is a central or most significant part of the trail's experience." -- [cite: 19, 69]

},

"estimated\_coverage\_percentage": {

"type": ["number", "null"],

"format": "float",

"minimum": 0,

"maximum": 100,

"description": "Approximate percentage (0-100) of the trail's total length that falls within this region." -- [cite: 21, 70]

},

"regional\_trail\_status": {

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

"enum": [

"fully\_operational",

"partially\_closed\_detours\_in\_place",

"seasonal\_access\_only",

"under\_development",

"closed",

"information\_unavailable",

null

],

"description": "Specific operational status of the trail within this region. Can override trails.operational\_status." -- [cite: 23, 71]

},

"regional\_status\_last\_checked": {

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

"format": "date-time",

"description": "Timestamp when regional\_trail\_status or notes were last actively verified/updated by a manager." -- [cite: 24, 72]

},

"regional\_status\_updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who last made a substantive update to regional\_trail\_status or regional\_significance\_notes." -- [cite: 25, 73]

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of record creation. Read-only.", -- [cite: 27]

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who created this trail-region link. Read-only.", -- [cite: 28]

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of last update for any field in this record. Read-only.", -- [cite: 30, 74]

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID (public.profiles.id) of the user who last made any update to this link record. Read-only.", -- [cite: 31]

"readOnly": true

}

},

"required": [

"trail\_id",

"region\_id",

"display\_order", -- [cite: 75]

"is\_entry\_point\_region",

"is\_exit\_point\_region",

"is\_primary\_focus\_region",

"created\_at",

"updated\_at"

],

"primary\_key": ["id"], -- [cite: 75]

"unique\_constraints": [

{"columns": ["trail\_id", "region\_id"], "name": "trail\_regions\_trail\_id\_region\_id\_key"}, -- [cite: 75]

{"columns": ["trail\_id", "display\_order"], "name": "trail\_regions\_trail\_id\_display\_order\_key"} -- [cite: 75]

]

}

```

5\. Relationships & Integrity

- Primary Key: Surrogate `id BIGINT GENERATED ALWAYS AS IDENTITY` is used for easier direct linking, particularly from the `public.translations` table.

- Unique Constraints:

- `UNIQUE (trail\_id, region\_id)`: Ensures a trail cannot be linked to the same region multiple times.

- `UNIQUE (trail\_id, display\_order)`: Ensures the sequence of regions for a given trail is unambiguous.

- Foreign Key Targets & `ON DELETE` Actions:

- `trail\_id` references `public.trails(id)`: `ON DELETE CASCADE`. If a trail is deleted, its associations with regions are also deleted.

- `region\_id` references `public.regions(id)`: `ON DELETE CASCADE`. If a region is deleted, its associations with trails are also deleted.

- `regional\_status\_updated\_by\_profile\_id`, `created\_by\_profile\_id`, `updated\_by\_profile\_id` reference `public.profiles(id)`: `ON DELETE SET NULL`.

- Mermaid ER Diagram Snippet:

Code snippet

```

erDiagram

trails {

bigint id PK

text name

}

regions {

bigint id PK

text name

}

profiles {

uuid id PK

text email

}

trail\_regions {

bigint id PK -- [cite: 9]

bigint trail\_id FK -- [cite: 11]

bigint region\_id FK -- [cite: 12]

integer display\_order -- [cite: 14]

text regional\_significance\_notes -- [cite: 15]

trail\_operational\_status\_enum regional\_trail\_status -- [cite: 23]

timestamptz regional\_status\_last\_checked -- [cite: 24]

uuid regional\_status\_updated\_by\_profile\_id FK -- [cite: 25]

timestamptz created\_at -- [cite: 27]

uuid created\_by\_profile\_id FK -- [cite: 28]

timestamptz updated\_at -- [cite: 30]

uuid updated\_by\_profile\_id FK -- [cite: 31]

-- UNIQUE (trail\_id, region\_id) [cite: 33]

-- UNIQUE (trail\_id, display\_order) [cite: 34]

}

translations { -- For context

bigint id PK

text table\_name

text column\_name

bigint row\_id FK -- References trail\_regions.id

text language\_code

text translated\_text

}

trails ||--|{ trail\_regions : "traverses" -- [cite: 80]

regions ||--|{ trail\_regions : "is\_traversed\_by" -- [cite: 81]

trail\_regions }o--|| profiles : "regional\_status\_updated\_by" -- [cite: 82, 88]

trail\_regions }o--|| profiles : "created\_by" -- [cite: 82, 89]

trail\_regions }o--|| profiles : "updated\_by" -- [cite: 82, 90]

trail\_regions ||..o| translations : "has\_translation\_for\_notes (via id)" -- [cite: 91]

```

6\. Multilingual Strategy

- Translatable Field: `regional\_significance\_notes`. The English base text is stored in this table.

- Linking Mechanism: Translations are managed in the central `public.translations` table. With the surrogate `id` PK on `trail\_regions`, linking is: `translations.table\_name = 'trail\_regions'`, `translations.column\_name = 'regional\_significance\_notes'`, `translations.row\_id = trail\_regions.id`.

7\. Role-Based Workflow & RLS Notes

- Workflow Fields: Includes standard audit columns (`created\_at`, `created\_by\_profile\_id`, `updated\_at`, `updated\_by\_profile\_id`) and specific fields for regional status management (`regional\_trail\_status`, `regional\_significance\_notes`, `regional\_status\_last\_checked`, `regional\_status\_updated\_by\_profile\_id`). The audit trigger `set\_trail\_region\_modification\_meta` handles these.

- 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.

- Row-Level Security (RLS) Policies (Conceptual Stubs):

- Public Read Access:

SQL

```

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

ON public.trail\_regions FOR SELECT

USING (

EXISTS (

SELECT 1 FROM public.trails t

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

AND t.content\_visibility\_status = 'published' -- Assuming 'published' status

AND t.deleted\_at IS NULL

)

); -- [cite: 98]

```

- Admin Full Access:

SQL

```

CREATE POLICY "Allow admin full access on trail\_regions"

ON public.trail\_regions FOR ALL

USING (public.is\_platform\_admin()) -- Assumes helper function

WITH CHECK (public.is\_platform\_admin()); -- [cite: 100]

```

- Regional Content Manager Access (Update):

SQL

```

CREATE POLICY "Allow regional managers to update their trail\_regions"

ON public.trail\_regions FOR UPDATE

USING (public.is\_regional\_manager\_for\_region(trail\_regions.region\_id)) -- Assumes helper function

WITH CHECK (public.is\_regional\_manager\_for\_region(trail\_regions.region\_id)); -- [cite: 101]

```

- Authorized User Access (Insert/Delete):

SQL

```

CREATE POLICY "Allow authorized users to insert trail\_regions"

ON public.trail\_regions FOR INSERT

WITH CHECK (public.is\_platform\_admin() OR public.can\_manage\_trail\_structure(trail\_regions.trail\_id)); -- Assumes helper functions [cite: 103]

CREATE POLICY "Allow authorized users to delete trail\_regions"

ON public.trail\_regions FOR DELETE

USING (public.is\_platform\_admin() OR public.can\_manage\_trail\_structure(trail\_regions.trail\_id)); -- Assumes helper functions [cite: 104]

```

8\. ENUM vs. Lookup Discussion

- `regional\_trail\_status` correctly reuses `public.trail\_operational\_status\_enum` from `trails` for consistency.

9\. UI/UX Enablement

- `display\_order` is crucial for correctly sequencing regions.

- `regional\_trail\_status` can drive UI indicators (e.g., warning icons).

- Boolean flags (`is\_entry\_point\_region`, etc.) can visually emphasize regions.

- `regional\_significance\_notes` provides context-specific information.

- `estimated\_coverage\_percentage` gives a sense of trail distribution.

- `regional\_status\_last\_checked` indicates data freshness.

10\. Key Considerations & Definitions

- Surrogate PK: The `id` column simplifies linking for translations.

- `display\_order` Integrity: Must be a positive, unique integer per `trail\_id` to define the trail's regional path.

- Status Hierarchy: `regional\_trail\_status` can override the general `trails.operational\_status`.

- Data Governance: Audit fields, especially `regional\_status\_last\_checked` and `regional\_status\_updated\_by\_profile\_id`, track the currency and authorship of regional status updates.

11\. Scalability & Future-Proofing

- `ON DELETE CASCADE` simplifies data management.

- Comprehensive audit columns support long-term maintenance.

- No soft deletes are needed as lifecycle is tied to parents.

- `BIGINT` FKs allow for growth.

- Future Enhancement: Full-Text Search (FTS) on `regional\_significance\_notes` could be considered as a future (Post-V2) enhancement if these notes become extensive and require advanced search capabilities, aligning with the project overview's future considerations.

12\. Next-Action Checklist

1\. 🔴 Implement DDL: Create/Update the `public.trail\_regions` table using the provided DDL. Ensure the surrogate `id` PK, `UNIQUE` constraints, all FKs with correct `ON DELETE` actions, audit/workflow columns, and the `set\_trail\_region\_modification\_meta` trigger are correctly defined and attached.

2\. 🔴 Implement Orphaned Translation Cleanup Trigger: Add an `AFTER DELETE` trigger on `public.trail\_regions` that calls `public.cleanup\_related\_translations('trail\_regions', OLD.id)` to remove orphaned translations for `regional\_significance\_notes` when a `trail\_regions` record is deleted.

3\. 🟠 Confirm Referenced Table PK Data Types: Verify that `public.regions.id` and `public.trails.id` primary keys are `BIGINT`. Adjust FK definitions in `trail\_regions` if necessary.

4\. 🟠 Integrate with `public.translations` Table: Ensure the `public.translations` table can effectively link to `trail\_regions` using `trail\_regions.id` as `row\_id` for the `regional\_significance\_notes` field.

5\. 🟠 Implement RLS Policies & Helper Functions: Define, implement, and thoroughly test the RLS policies. This includes creating helper functions like `public.is\_platform\_admin()`, `public.is\_regional\_manager\_for\_region(BIGINT)`, and `public.can\_manage\_trail\_structure(BIGINT)`.

6\. 🟢 Data Migration (if applicable): If migrating from an older structure, plan for the addition of the surrogate `id`. Populate new audit columns for existing rows as feasible.

\* \* \* \* \*

### Diff Summary

- Schema/DDL:

- Confirmed use of surrogate `id BIGINT PRIMARY KEY` as per the original document's reconstruction.

- Standardized comments with "Version: V2." and ensured full schema paths (e.g., `public.trails(id)`).

- Verified `ON DELETE CASCADE` for `trail\_id` and `region\_id` FKs.

- Ensured audit trigger `set\_trail\_region\_modification\_meta` logic is sound and includes setting `updated\_by\_profile\_id` on all operations and specific logic for `regional\_status\_updated\_by\_profile\_id` and `regional\_status\_last\_checked`.

- JSON Schema Mirror:

- Updated descriptions to include "Version: V2." and reference full table paths for FKs.

- Marked `readOnly` for auto-generated/audit fields.

- Added `is\_entry\_point\_region`, `is\_exit\_point\_region`, `is\_primary\_focus\_region`, `created\_at`, `updated\_at` to the `required` array based on `NOT NULL` constraints with defaults.

- Multilingual Strategy:

- Clarified that `regional\_significance\_notes` is the translatable field and the linking uses the new surrogate `id`.

- RLS Policies:

- Provided more explicit SQL stubs, referencing necessary helper functions.

- Scalability & Future-Proofing:

- Added a note about considering Full-Text Search (FTS) for `regional\_significance\_notes` as a future enhancement.

- Next-Action Checklist:

- 🔴 Added the critical action: "Implement Orphaned Translation Cleanup Trigger."

- Added severity icons (🔴🟠🟢) to checklist items.

- Emphasized creation and testing of RLS helper functions.