<https://gemini.google.com/u/1/app/42a77d08d7a45bcb>

**Database Views Specification - Module 7: Curated Itinerary Module (V2.1)**

**1. Introduction**

This document outlines the specifications for database views created to support the "Curated Itinerary Module" (Module 7) of the Via di Francesco Pilgrimage Platform V2.1. Views are virtual tables whose contents are defined by a query. They are used to simplify complex queries, encapsulate logic, improve read performance for common data aggregations, and provide a stable interface for applications and APIs, even if underlying table structures change. Each view specification will include its name, purpose, DDL, key columns, and usage notes.

**2. Module 7: Curated Itinerary Module Views**

These views are designed to support common API queries related to curated itineraries, providing localized and aggregated information.

**2.1. View:** public.v\_curated\_itineraries\_list\_localized

* **Purpose**: To provide a summarized, localized list of curated itineraries suitable for Browse, filtering, and display in list UIs. Each row represents an itinerary in a specific language, including key translated fields and references to related master data.
* **DDL**:
* SQL

CREATE OR REPLACE VIEW public.v\_curated\_itineraries\_list\_localized AS

SELECT

ci.id AS itinerary\_id,

ci.slug AS itinerary\_slug,

ci.is\_featured\_itinerary,

ci.total\_walking\_days\_approx,

ci.total\_distance\_km\_approx,

ci.content\_status\_id, -- For filtering by status ID

cs\_master.code AS content\_status\_code, -- For displaying status code

tdl\_master.code AS difficulty\_level\_code, -- For displaying difficulty code

ci.banner\_image\_media\_id, -- For fetching a thumbnail via media table

-- Primarily translated fields for the list

name\_trans.language\_code,

COALESCE(name\_trans.translated\_text, ci.name) AS name, -- Fallback to base name

COALESCE(short\_desc\_trans.translated\_text, ci.short\_description) AS short\_description, -- Fallback to base short\_description

COALESCE(cs\_name\_trans.translated\_text, cs\_master\_base.name) AS content\_status\_name, -- Translated status name

COALESCE(tdl\_name\_trans.translated\_text, tdl\_master\_base.name) AS difficulty\_level\_name, -- Translated difficulty name

-- Aggregated category and season codes for filtering/display - names would require more complex joins/subqueries or API-level fetching

(SELECT array\_agg(icm.category\_code)

FROM public.curated\_itinerary\_to\_category citc

JOIN public.itinerary\_categories\_master icm ON citc.category\_id = icm.id

WHERE citc.itinerary\_id = ci.id AND icm.is\_active = true) AS category\_codes,

(SELECT array\_agg(sm.season\_code)

FROM public.curated\_itinerary\_to\_season cits

JOIN public.seasons\_master sm ON cits.season\_id = sm.id

WHERE cits.itinerary\_id = ci.id AND sm.is\_active = true) AS season\_codes,

ci.created\_at AS itinerary\_created\_at,

ci.updated\_at AS itinerary\_updated\_at,

ci.deleted\_at AS itinerary\_deleted\_at -- For filtering out deleted items

FROM

public.curated\_itineraries ci

-- Join for Itinerary Name (drives the language for this view row)

LEFT JOIN public.translations name\_trans

ON name\_trans.table\_identifier = 'curated\_itineraries'

AND name\_trans.row\_foreign\_key = ci.id::TEXT

AND name\_trans.column\_identifier = 'name'

AND name\_trans.translation\_status = 'published\_live' -- Assuming this status

-- Join for Itinerary Short Description

LEFT JOIN public.translations short\_desc\_trans

ON short\_desc\_trans.table\_identifier = 'curated\_itineraries'

AND short\_desc\_trans.row\_foreign\_key = ci.id::TEXT

AND short\_desc\_trans.column\_identifier = 'short\_description'

AND short\_desc\_trans.language\_code = name\_trans.language\_code -- Match language

AND short\_desc\_trans.translation\_status = 'published\_live'

-- Join for Content Status Master (to get base name and code)

LEFT JOIN public.content\_statuses\_master cs\_master ON ci.content\_status\_id = cs\_master.id

LEFT JOIN public.content\_statuses\_master cs\_master\_base ON ci.content\_status\_id = cs\_master\_base.id -- Separate alias for base name

LEFT JOIN public.translations cs\_name\_trans

ON cs\_name\_trans.table\_identifier = 'content\_statuses\_master'

AND cs\_name\_trans.row\_foreign\_key = cs\_master.id::TEXT

AND cs\_name\_trans.column\_identifier = 'name' -- Assuming 'name' is the translatable field in content\_statuses\_master

AND cs\_name\_trans.language\_code = name\_trans.language\_code

AND cs\_name\_trans.translation\_status = 'published\_live'

-- Join for Trail Difficulty Levels Master (to get base name and code)

LEFT JOIN public.trail\_difficulty\_levels\_master tdl\_master ON ci.overall\_difficulty\_level\_id = tdl\_master.id

LEFT JOIN public.trail\_difficulty\_levels\_master tdl\_master\_base ON ci.overall\_difficulty\_level\_id = tdl\_master\_base.id -- Separate alias for base name

LEFT JOIN public.translations tdl\_name\_trans

ON tdl\_name\_trans.table\_identifier = 'trail\_difficulty\_levels\_master'

AND tdl\_name\_trans.row\_foreign\_key = tdl\_master.id::TEXT

AND tdl\_name\_trans.column\_identifier = 'name' -- Assuming 'name' is translatable in trail\_difficulty\_levels\_master

AND tdl\_name\_trans.language\_code = name\_trans.language\_code

AND tdl\_name\_trans.translation\_status = 'published\_live'

WHERE

ci.deleted\_at IS NULL; -- Typically, RLS policies would further filter by content\_status\_id

COMMENT ON VIEW public.v\_curated\_itineraries\_list\_localized IS 'Provides a summarized, localized list of curated itineraries with key identifying information, statistics, status, difficulty, and aggregated category/season codes. Version 2.1.';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.name IS 'Localized name of the curated itinerary.';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.short\_description IS 'Localized short description of the curated itinerary.';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.content\_status\_name IS 'Localized name of the content status.';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.difficulty\_level\_name IS 'Localized name of the difficulty level.';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.category\_codes IS 'Array of category codes associated with the itinerary (from active categories).';

COMMENT ON COLUMN public.v\_curated\_itineraries\_list\_localized.season\_codes IS 'Array of season codes associated with the itinerary (from active seasons).';

* **Key Columns**:
  + itinerary\_id: BIGINT - Unique ID of the itinerary.
  + language\_code: TEXT - Language code for the localized fields in this row.
  + name: TEXT - Localized name of the itinerary.
  + slug: TEXT - URL-friendly slug.
  + short\_description: TEXT - Localized short description.
  + banner\_image\_media\_id: UUID - ID of the banner media item (for thumbnail lookup).
  + total\_distance\_km\_approx: REAL - Approximate total distance.
  + total\_walking\_days\_approx: INTEGER - Approximate walking days.
  + difficulty\_level\_code: TEXT - Code of the difficulty level.
  + difficulty\_level\_name: TEXT - Localized name of the difficulty level.
  + content\_status\_code: TEXT - Code of the content status.
  + content\_status\_name: TEXT - Localized name of the content status.
  + category\_codes: TEXT[] - Array of category\_codes from itinerary\_categories\_master.
  + season\_codes: TEXT[] - Array of season\_codes from seasons\_master.
  + is\_featured\_itinerary: BOOLEAN - Flag if itinerary is featured.
  + itinerary\_deleted\_at: TIMESTAMPTZ - For filtering soft-deleted items.
* **Usage Notes**:
  + This view is intended for read-only operations.
  + RLS policies on curated\_itineraries, translations, and master tables will apply. API queries should filter by content\_status\_code = 'PUBLISHED' (or equivalent) for public listings.
  + Performance relies on good indexing on curated\_itineraries (PK, content\_status\_id, overall\_difficulty\_level\_id, deleted\_at), translations (composite index on table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translation\_status), and the master/junction tables used in subqueries.
  + The language\_code in the output is primarily driven by the translation of curated\_itineraries.name. If a translation for the name doesn't exist for a requested language, that itinerary might not appear for that language via this view's logic, or a fallback mechanism using COALESCE with the base language from curated\_itineraries itself is used.
  + Fetching translated names for category\_codes and season\_codes within this list view (rather than just codes) would add significant complexity (e.g., more joins, array manipulations, or JSON functions); it's often more efficient for the API to fetch codes here and then fetch their translations separately if needed for filter UIs, or for the client to map codes to translated names from a pre-fetched list of all categories/seasons.

**2.2. View:** public.v\_curated\_itinerary\_detail\_localized

* **Purpose**: To provide a comprehensive, denormalized, and localized representation of a single curated itinerary's attributes. This includes its direct translatable fields, linked master data (difficulty, status with their translated names), primary media details (with translated alt text), and aggregated categories & seasons.
* **DDL**:
* SQL

CREATE OR REPLACE VIEW public.v\_curated\_itinerary\_detail\_localized AS

WITH localized\_categories AS (

SELECT

citc.itinerary\_id,

name\_trans.language\_code,

jsonb\_agg(DISTINCT jsonb\_build\_object(

'category\_code', icm.category\_code,

'name', COALESCE(cat\_name\_trans.translated\_text, icm.name), -- Assuming icm.name is base lang

'description', COALESCE(cat\_desc\_trans.translated\_text, icm.description), -- Assuming icm.description is base lang

'icon\_identifier', icm.icon\_identifier

) ORDER BY icm.sort\_order, icm.name) AS categories\_jsonb

FROM public.curated\_itinerary\_to\_category citc

JOIN public.itinerary\_categories\_master icm ON citc.category\_id = icm.id AND icm.is\_active = true

JOIN public.translations name\_trans -- This join determines the overall language context from itinerary name

ON name\_trans.table\_identifier = 'curated\_itineraries'

AND name\_trans.row\_foreign\_key = citc.itinerary\_id::TEXT -- links to the itinerary ID

AND name\_trans.column\_identifier = 'name'

AND name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations cat\_name\_trans

ON cat\_name\_trans.table\_identifier = 'itinerary\_categories\_master'

AND cat\_name\_trans.row\_foreign\_key = icm.id::TEXT

AND cat\_name\_trans.column\_identifier = 'name'

AND cat\_name\_trans.language\_code = name\_trans.language\_code

AND cat\_name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations cat\_desc\_trans

ON cat\_desc\_trans.table\_identifier = 'itinerary\_categories\_master'

AND cat\_desc\_trans.row\_foreign\_key = icm.id::TEXT

AND cat\_desc\_trans.column\_identifier = 'description'

AND cat\_desc\_trans.language\_code = name\_trans.language\_code

AND cat\_desc\_trans.translation\_status = 'published\_live'

GROUP BY citc.itinerary\_id, name\_trans.language\_code

), localized\_seasons AS (

SELECT

cits.itinerary\_id,

name\_trans.language\_code,

jsonb\_agg(DISTINCT jsonb\_build\_object(

'season\_code', sm.season\_code,

'name', COALESCE(season\_name\_trans.translated\_text, sm.name), -- Assuming sm.name is base lang

'description', COALESCE(season\_desc\_trans.translated\_text, sm.description), -- Assuming sm.description is base lang

'icon\_identifier', sm.icon\_identifier

) ORDER BY sm.sort\_order, sm.name) AS seasons\_jsonb

FROM public.curated\_itinerary\_to\_season cits

JOIN public.seasons\_master sm ON cits.season\_id = sm.id AND sm.is\_active = true

JOIN public.translations name\_trans -- This join determines the overall language context from itinerary name

ON name\_trans.table\_identifier = 'curated\_itineraries'

AND name\_trans.row\_foreign\_key = cits.itinerary\_id::TEXT -- links to the itinerary ID

AND name\_trans.column\_identifier = 'name'

AND name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations season\_name\_trans

ON season\_name\_trans.table\_identifier = 'seasons\_master'

AND season\_name\_trans.row\_foreign\_key = sm.id::TEXT

AND season\_name\_trans.column\_identifier = 'name'

AND season\_name\_trans.language\_code = name\_trans.language\_code

AND season\_name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations season\_desc\_trans

ON season\_desc\_trans.table\_identifier = 'seasons\_master'

AND season\_desc\_trans.row\_foreign\_key = sm.id::TEXT

AND season\_desc\_trans.column\_identifier = 'description'

AND season\_desc\_trans.language\_code = name\_trans.language\_code

AND season\_desc\_trans.translation\_status = 'published\_live'

GROUP BY cits.itinerary\_id, name\_trans.language\_code

)

SELECT

ci.id AS itinerary\_id,

ci.slug AS itinerary\_slug,

ci.itinerary\_code,

ci.primary\_trail\_id, -- Keep FKs for potential client-side linking or further API calls

ci.based\_on\_primary\_route\_id,

ci.total\_walking\_days\_approx,

ci.total\_nights\_approx,

ci.total\_distance\_km\_approx,

ci.overall\_difficulty\_level\_id,

ci.start\_waypoint\_id\_approx,

ci.end\_waypoint\_id\_approx,

ci.content\_status\_id,

ci.published\_at,

ci.is\_featured\_itinerary,

ci.version\_notes,

ci.summary\_stats\_last\_calculated\_at,

ci.data\_last\_verified\_at,

ci.created\_at AS itinerary\_created\_at,

ci.updated\_at AS itinerary\_updated\_at,

ci.deleted\_at AS itinerary\_deleted\_at,

creator\_profile.public\_display\_name AS created\_by\_display\_name,

updater\_profile.public\_display\_name AS updated\_by\_display\_name,

-- Language of this particular localized record

name\_trans.language\_code,

-- Translated Curated Itinerary Fields

COALESCE(name\_trans.translated\_text, ci.name) AS name,

COALESCE(short\_desc\_trans.translated\_text, ci.short\_description) AS short\_description,

COALESCE(long\_desc\_trans.translated\_text, ci.long\_description) AS long\_description,

COALESCE(theme\_focus\_trans.translated\_text, ci.theme\_or\_focus) AS theme\_or\_focus,

COALESCE(suitability\_notes\_trans.translated\_text, ci.suitability\_notes) AS suitability\_notes,

COALESCE(start\_loc\_trans.translated\_text, ci.primary\_start\_location\_text) AS primary\_start\_location\_text,

COALESCE(end\_loc\_trans.translated\_text, ci.primary\_end\_location\_text) AS primary\_end\_location\_text,

-- Difficulty Level Details (Localized)

tdl\_master.code AS difficulty\_level\_code,

COALESCE(tdl\_name\_trans.translated\_text, tdl\_master\_base.name) AS difficulty\_level\_name,

COALESCE(tdl\_desc\_trans.translated\_text, tdl\_master\_base.description) AS difficulty\_level\_description,

tdl\_master.icon\_identifier AS difficulty\_level\_icon,

-- Content Status Details (Localized)

cs\_master.code AS content\_status\_code,

COALESCE(cs\_name\_trans.translated\_text, cs\_master\_base.name) AS content\_status\_name,

COALESCE(cs\_desc\_trans.translated\_text, cs\_master\_base.description) AS content\_status\_description,

-- Banner Image Details (Localized Alt Text)

banner\_media.id AS banner\_media\_id,

banner\_media.storage\_object\_path\_original AS banner\_media\_original\_url,

banner\_media.image\_variants\_json AS banner\_media\_variants,

COALESCE(banner\_alt\_trans.translated\_text, banner\_media.default\_alt\_text) AS banner\_media\_alt\_text, -- Assuming default\_alt\_text in media table

-- Map Overview Image Details (Localized Alt Text)

map\_media.id AS map\_overview\_media\_id,

map\_media.storage\_object\_path\_original AS map\_overview\_media\_original\_url,

map\_media.image\_variants\_json AS map\_overview\_media\_variants,

COALESCE(map\_alt\_trans.translated\_text, map\_media.default\_alt\_text) AS map\_overview\_media\_alt\_text,

-- Aggregated Categories (Localized)

lc.categories\_jsonb,

-- Aggregated Seasons (Localized)

ls.seasons\_jsonb

FROM

public.curated\_itineraries ci

-- Profile Joins

LEFT JOIN public.profiles creator\_profile ON ci.created\_by\_profile\_id = creator\_profile.id

LEFT JOIN public.profiles updater\_profile ON ci.updated\_by\_profile\_id = updater\_profile.id

-- Join for Itinerary Name (drives the language for this view row)

INNER JOIN public.translations name\_trans

ON name\_trans.table\_identifier = 'curated\_itineraries'

AND name\_trans.row\_foreign\_key = ci.id::TEXT

AND name\_trans.column\_identifier = 'name'

AND name\_trans.translation\_status = 'published\_live' -- Assuming this status

-- Left Joins for other translatable fields of the itinerary entity itself

LEFT JOIN public.translations short\_desc\_trans ON short\_desc\_trans.table\_identifier = 'curated\_itineraries' AND short\_desc\_trans.row\_foreign\_key = ci.id::TEXT AND short\_desc\_trans.column\_identifier = 'short\_description' AND short\_desc\_trans.language\_code = name\_trans.language\_code AND short\_desc\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations long\_desc\_trans ON long\_desc\_trans.table\_identifier = 'curated\_itineraries' AND long\_desc\_trans.row\_foreign\_key = ci.id::TEXT AND long\_desc\_trans.column\_identifier = 'long\_description' AND long\_desc\_trans.language\_code = name\_trans.language\_code AND long\_desc\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations theme\_focus\_trans ON theme\_focus\_trans.table\_identifier = 'curated\_itineraries' AND theme\_focus\_trans.row\_foreign\_key = ci.id::TEXT AND theme\_focus\_trans.column\_identifier = 'theme\_or\_focus' AND theme\_focus\_trans.language\_code = name\_trans.language\_code AND theme\_focus\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations suitability\_notes\_trans ON suitability\_notes\_trans.table\_identifier = 'curated\_itineraries' AND suitability\_notes\_trans.row\_foreign\_key = ci.id::TEXT AND suitability\_notes\_trans.column\_identifier = 'suitability\_notes' AND suitability\_notes\_trans.language\_code = name\_trans.language\_code AND suitability\_notes\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations start\_loc\_trans ON start\_loc\_trans.table\_identifier = 'curated\_itineraries' AND start\_loc\_trans.row\_foreign\_key = ci.id::TEXT AND start\_loc\_trans.column\_identifier = 'primary\_start\_location\_text' AND start\_loc\_trans.language\_code = name\_trans.language\_code AND start\_loc\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations end\_loc\_trans ON end\_loc\_trans.table\_identifier = 'curated\_itineraries' AND end\_loc\_trans.row\_foreign\_key = ci.id::TEXT AND end\_loc\_trans.column\_identifier = 'primary\_end\_location\_text' AND end\_loc\_trans.language\_code = name\_trans.language\_code AND end\_loc\_trans.translation\_status = 'published\_live'

-- Join for Difficulty Level Master and its translations

LEFT JOIN public.trail\_difficulty\_levels\_master tdl\_master ON ci.overall\_difficulty\_level\_id = tdl\_master.id

LEFT JOIN public.trail\_difficulty\_levels\_master tdl\_master\_base ON ci.overall\_difficulty\_level\_id = tdl\_master\_base.id

LEFT JOIN public.translations tdl\_name\_trans ON tdl\_name\_trans.table\_identifier = 'trail\_difficulty\_levels\_master' AND tdl\_name\_trans.row\_foreign\_key = tdl\_master.id::TEXT AND tdl\_name\_trans.column\_identifier = 'name' AND tdl\_name\_trans.language\_code = name\_trans.language\_code AND tdl\_name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations tdl\_desc\_trans ON tdl\_desc\_trans.table\_identifier = 'trail\_difficulty\_levels\_master' AND tdl\_desc\_trans.row\_foreign\_key = tdl\_master.id::TEXT AND tdl\_desc\_trans.column\_identifier = 'description' AND tdl\_desc\_trans.language\_code = name\_trans.language\_code AND tdl\_desc\_trans.translation\_status = 'published\_live'

-- Join for Content Status Master and its translations

LEFT JOIN public.content\_statuses\_master cs\_master ON ci.content\_status\_id = cs\_master.id

LEFT JOIN public.content\_statuses\_master cs\_master\_base ON ci.content\_status\_id = cs\_master\_base.id

LEFT JOIN public.translations cs\_name\_trans ON cs\_name\_trans.table\_identifier = 'content\_statuses\_master' AND cs\_name\_trans.row\_foreign\_key = cs\_master.id::TEXT AND cs\_name\_trans.column\_identifier = 'name' AND cs\_name\_trans.language\_code = name\_trans.language\_code AND cs\_name\_trans.translation\_status = 'published\_live'

LEFT JOIN public.translations cs\_desc\_trans ON cs\_desc\_trans.table\_identifier = 'content\_statuses\_master' AND cs\_desc\_trans.row\_foreign\_key = cs\_master.id::TEXT AND cs\_desc\_trans.column\_identifier = 'description' AND cs\_desc\_trans.language\_code = name\_trans.language\_code AND cs\_desc\_trans.translation\_status = 'published\_live'

-- Join for Banner Media and its alt text translation

LEFT JOIN public.media banner\_media ON ci.banner\_image\_media\_id = banner\_media.id

LEFT JOIN public.translations banner\_alt\_trans ON banner\_alt\_trans.table\_identifier = 'media' AND banner\_alt\_trans.row\_foreign\_key = banner\_media.id::TEXT AND banner\_alt\_trans.column\_identifier = 'default\_alt\_text' AND banner\_alt\_trans.language\_code = name\_trans.language\_code AND banner\_alt\_trans.translation\_status = 'published\_live'

-- Join for Map Overview Media and its alt text translation

LEFT JOIN public.media map\_media ON ci.map\_overview\_image\_media\_id = map\_media.id

LEFT JOIN public.translations map\_alt\_trans ON map\_alt\_trans.table\_identifier = 'media' AND map\_alt\_trans.row\_foreign\_key = map\_media.id::TEXT AND map\_alt\_trans.column\_identifier = 'default\_alt\_text' AND map\_alt\_trans.language\_code = name\_trans.language\_code AND map\_alt\_trans.translation\_status = 'published\_live'

-- Join CTEs for aggregated categories and seasons

LEFT JOIN localized\_categories lc ON ci.id = lc.itinerary\_id AND name\_trans.language\_code = lc.language\_code

LEFT JOIN localized\_seasons ls ON ci.id = ls.itinerary\_id AND name\_trans.language\_code = ls.language\_code

WHERE

ci.deleted\_at IS NULL;

COMMENT ON VIEW public.v\_curated\_itinerary\_detail\_localized IS 'Provides comprehensive, denormalized, and localized details for a single curated itinerary, including direct attributes, linked master data (difficulty, status), primary media, and aggregated categories/seasons. Each row represents an itinerary in a specific language. Segments are fetched separately. Version 2.1.';

* **Key Columns (Abridged)**:
  + itinerary\_id: BIGINT - Unique ID of the itinerary.
  + language\_code: TEXT - Language code for the localized fields.
  + name: TEXT - Localized name.
  + slug: TEXT.
  + long\_description: TEXT - Localized long description.
  + difficulty\_level\_name: TEXT - Localized difficulty name.
  + content\_status\_name: TEXT - Localized status name.
  + banner\_media\_variants: JSONB - Variants for banner image.
  + banner\_media\_alt\_text: TEXT - Localized alt text for banner.
  + map\_overview\_media\_variants: JSONB - Variants for map image.
  + map\_overview\_media\_alt\_text: TEXT - Localized alt text for map.
  + categories\_jsonb: JSONB - Array of localized category objects {category\_code, name, description, icon\_identifier}.
  + seasons\_jsonb: JSONB - Array of localized season objects {season\_code, name, description, icon\_identifier}.
  + Plus all other relevant fields from curated\_itineraries and linked master tables.
* **Example Usage**:
* SQL

SELECT

itinerary\_id,

name, -- Localized name

long\_description, -- Localized description

difficulty\_level\_name, -- Localized

categories\_jsonb,

seasons\_jsonb,

banner\_media\_variants ->> 'display\_l\_jpeg' AS banner\_url

FROM

public.v\_curated\_itinerary\_detail\_localized

WHERE

itinerary\_id = 123 AND language\_code = 'it'

AND content\_status\_code = 'PUBLISHED'; -- Further filtered by API or RLS

* **Underlying Tables & Key Joins**:
  + curated\_itineraries (Primary)
  + public.translations (Multiple LEFT JOINs for itinerary fields, master data fields, media alt text; INNER JOIN for primary itinerary name).
  + trail\_difficulty\_levels\_master, content\_statuses\_master, itinerary\_categories\_master (via CTE), seasons\_master (via CTE).
  + public.media (for banner, map images).
  + public.profiles (for creator/updater display names).
* **RLS (Row-Level Security) Considerations**:
  + View should be defined with SECURITY INVOKER (default).
  + RLS policies on underlying tables (curated\_itineraries, translations, media, etc.) will apply.
  + API queries should additionally filter by content\_status\_code = 'PUBLISHED' for public access, or rely on RLS on curated\_itineraries which should already incorporate this.
* **Performance & Optimization Notes**:
  + Indexing on public.translations (composite index on table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translation\_status) is critical.
  + Indexes on all FKs and commonly filtered fields in curated\_itineraries and master tables are essential.
  + The CTEs for categories and seasons aim to pre-aggregate data to avoid excessive joins in the final select. Performance should be tested.
  + Consider materializing this view if read performance for frequently accessed, complex itinerary details becomes a bottleneck.
* **Assumptions & Dependencies**:
  + Assumes the structure of public.translations, public.media (with default\_alt\_text, image\_variants\_json), and master tables.
  + Assumes translation\_status = 'published\_live' for production-ready translations.
  + Translatable fields in master tables like itinerary\_categories\_master.name are indeed translated via the translations table.
  + The COALESCE pattern provides a fallback to the base language content stored directly in curated\_itineraries or master tables if a translation is not found for the requested language\_code.
* **Next-Action Checklist**:
  + 🔴 Finalize all translatable field identifiers for curated\_itineraries, media, and all involved master tables used in translations.
  + 🟠 Implement and thoroughly test the DDL for v\_curated\_itineraries\_list\_localized and v\_curated\_itinerary\_detail\_localized.
  + 🟠 Test query performance extensively, especially for v\_curated\_itinerary\_detail\_localized due to its complexity.
  + 🟢 Optimize underlying table indexes based on view query plans.
  + 🟢 Document these views for API developers, including output columns, query patterns, and how related data (like segments) should be fetched.

This provides a starting point for creating production-ready views for Module 7. The exact implementation of JSON aggregation and CTEs might need adjustments based on your PostgreSQL version and specific performance characteristics.