<https://gemini.google.com/u/1/app/b76d14705d5101f9?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/42a77d08d7a45bcb

\* \* \* \* \*

Production-Ready Specification: public.curated\_itinerary\_to\_category Table

Version: 2.1 (V2.1 Checklist REV 05-18-25-A Applied)

Date: May 18, 2025

1\. Purpose & Primary Use-Cases

Establishes a many-to-many relationship between curated itineraries (`curated\_itineraries`) and itinerary categories (`itinerary\_categories\_master`). Each row signifies that a specific itinerary is associated with a particular category. This table enables filtering and classification of itineraries.

\*Key User-Story Touchpoints:\*

- Pilgrim (Anna): Enables filtering itineraries by category (A7).

- Platform Administrator (Admin Team) / Regional Content Manager (Sofia): Allows assignment of multiple categories to itineraries (D2).

2\. Schema

| Column | Data Type | Constraints | Description |

| `itinerary\_id` | `bigint` | Primary Key (Composite), Foreign Key to `curated\_itineraries(id)` ON DELETE CASCADE, Not Null | Identifier of the curated itinerary. |

| `category\_id` | `integer` | Primary Key (Composite), Foreign Key to `itinerary\_categories\_master(id)` ON DELETE RESTRICT, Not Null | Identifier of the itinerary category. |

| `created\_at` | `timestamp with time zone` | Not Null, Default `now()` | Timestamp when the association was created. |

| `created\_by\_profile\_id` | `uuid` | Foreign Key to `public.profiles(id)` ON DELETE SET NULL, Nullable | Profile ID of the user who created this link. |

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

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

3\. PostgreSQL DDL

SQL

```

CREATE TABLE public.curated\_itinerary\_to\_category (

itinerary\_id BIGINT NOT NULL REFERENCES public.curated\_itineraries(id) ON DELETE CASCADE,

category\_id INTEGER NOT NULL REFERENCES public.itinerary\_categories\_master(id) ON DELETE RESTRICT,

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

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

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

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

PRIMARY KEY (itinerary\_id, category\_id)

);

-- Indexes

-- Composite PK creates a unique index on (itinerary\_id, category\_id).

CREATE INDEX IF NOT EXISTS idx\_curated\_itinerary\_to\_category\_itinerary\_id ON public.curated\_itinerary\_to\_category(itinerary\_id);

CREATE INDEX IF NOT EXISTS idx\_curated\_itinerary\_to\_category\_category\_id ON public.curated\_itinerary\_to\_category(category\_id);

CREATE INDEX IF NOT EXISTS idx\_curated\_itinerary\_to\_category\_created\_by ON public.curated\_itinerary\_to\_category(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_curated\_itinerary\_to\_category\_updated\_by ON public.curated\_itinerary\_to\_category(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

-- Comments

COMMENT ON TABLE public.curated\_itinerary\_to\_category IS 'Junction table linking curated itineraries to their assigned categories (Many-to-Many relationship). Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.itinerary\_id IS 'Foreign key to the public.curated\_itineraries table. Part of composite PK. ON DELETE CASCADE. Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.category\_id IS 'Foreign key to the public.itinerary\_categories\_master table. Part of composite PK. ON DELETE RESTRICT. Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.created\_at IS 'Timestamp when the itinerary-category association was made. Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.created\_by\_profile\_id IS 'Profile ID of the user who created this link. FK to public.profiles.id. ON DELETE SET NULL. Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.updated\_at IS 'Timestamp when this link record was last updated (auto-updated by trigger). Version 2.1.';

COMMENT ON COLUMN public.curated\_itinerary\_to\_category.updated\_by\_profile\_id IS 'Profile ID of the user who last updated this link. FK to public.profiles.id. ON DELETE SET NULL. Version 2.1.';

```

4\. Triggers/Functions

- Audit Metadata Trigger:

SQL

```

-- Assuming a generic public.set\_linking\_table\_audit\_meta() or public.set\_master\_table\_audit\_meta() function is defined

-- that handles created\_at, updated\_at, created\_by\_profile\_id, and updated\_by\_profile\_id.

-- Example:

-- CREATE OR REPLACE FUNCTION public.set\_linking\_table\_audit\_meta()... (as defined previously)

CREATE TRIGGER trigger\_set\_curated\_itinerary\_to\_category\_audit\_meta

BEFORE INSERT OR UPDATE ON public.curated\_itinerary\_to\_category

FOR EACH ROW

EXECUTE FUNCTION public.set\_linking\_table\_audit\_meta(); -- Or a more generic audit trigger

```

\*Comment on Trigger\*: Manages `created\_at`, `updated\_at`, `created\_by\_profile\_id`, and `updated\_by\_profile\_id` columns.

5\. JSON Schema Mirror

JSON

```

{

"title": "curated\_itinerary\_to\_category",

"description": "Junction table linking curated itineraries to their assigned categories (Many-to-Many relationship). Version 2.1.",

"type": "object",

"properties": {

"itinerary\_id": {

"type": "integer",

"format": "int64",

"description": "Identifier of the curated itinerary. FK to public.curated\_itineraries.id. Part of composite PK. Version 2.1."

},

"category\_id": {

"type": "integer",

"description": "Identifier of the itinerary category. FK to public.itinerary\_categories\_master.id. Part of composite PK. Version 2.1."

},

"created\_at": {

"type": "string",

"format": "date-time",

"readOnly": true,

"description": "Timestamp when the association was created. Read-only. Version 2.1."

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who created this link. FK to public.profiles.id. Read-only. Version 2.1.",

"readOnly": true

},

"updated\_at": {

"type": "string",

"format": "date-time",

"readOnly": true,

"description": "Timestamp when this link record was last updated (auto-updated by trigger). Read-only. Version 2.1."

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who last updated this link. FK to public.profiles.id. Read-only. Version 2.1.",

"readOnly": true

}

},

"required": [

"itinerary\_id",

"category\_id",

"created\_at",

"updated\_at"

],

"primary\_key": ["itinerary\_id", "category\_id"]

}

```

6\. Relationships & Integrity

- Composite Primary Key: `(itinerary\_id, category\_id)`.

- `itinerary\_id` -> `public.curated\_itineraries(id)`: `ON DELETE CASCADE`.

- `category\_id` -> `public.itinerary\_categories\_master(id)`: `ON DELETE RESTRICT`.

- Audit FKs to `public.profiles(id)`.

- Mermaid ER Snippet:

Code snippet

```

erDiagram

curated\_itineraries {

bigint id PK

}

itinerary\_categories\_master {

integer id PK

}

profiles {

uuid id PK

}

curated\_itinerary\_to\_category {

bigint itinerary\_id PK FK

integer category\_id PK FK

timestamptz created\_at

uuid created\_by\_profile\_id FK

timestamptz updated\_at

uuid updated\_by\_profile\_id FK

}

curated\_itineraries ||--|{ curated\_itinerary\_to\_category : "has categories (CASCADE)"

itinerary\_categories\_master ||--|{ curated\_itinerary\_to\_category : "categorizes itineraries (RESTRICT)"

curated\_itinerary\_to\_category }o--|| profiles : "created\_by (SET NULL)"

curated\_itinerary\_to\_category }o--|| profiles : "updated\_by (SET NULL)"

```

7\. Multilingual Strategy

- ⚪ N/A - This table contains no direct translatable text. It links records that have their own multilingual strategies.

8\. Role-Based Workflow & RLS Notes

- Workflow: `created\_at` and other audit columns track link creation/modification.

- RLS Policies (Conceptual): Aligned with parent `curated\_itineraries` permissions. If a user can edit an itinerary, they can manage its category links.

9\. ENUM vs Lookup Discussion

- ⚪ N/A - This junction table supports the `itinerary\_categories\_master` lookup table.

10\. UI/UX Enablement

- Fundamental for assigning categories to itineraries and enabling category-based filtering.

- PK and FK indexes support query performance.

11\. Auditing & Lifecycle Management

- Audit Columns: `created\_at`, `updated\_at`, `created\_by\_profile\_id`, `updated\_by\_profile\_id`.

- Lifecycle: No soft delete. Rows are hard-deleted when a link is removed or parent itinerary is deleted (CASCADE).

12\. Scalability & Future-Proofing

- Standard junction table design, scales with number of itineraries and average categories per itinerary.

13\. Seed Data

- ⚪ N/A - Transactional linking data.

14\. Next-Action Checklist

- 🔴 Implement DDL: Execute `CREATE TABLE` for `public.curated\_itinerary\_to\_category` with all audit columns, PK, FKs, and indexes.

- 🔴 Implement Audit Trigger: Apply `public.set\_linking\_table\_audit\_meta` (or similar).

- 🟠 Verify FK Data Types: Ensure `itinerary\_id` is `bigint` and `category\_id` is `integer`.

- 🟠 Implement RLS Policies.

- 🟢 Test integrity rules (`ON DELETE CASCADE`/`RESTRICT`).

- 🟢 Update Admin UI for category assignment.

\* \* \* \* \*