<https://gemini.google.com/u/1/app/bf28f389cd093e55?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/0e3ccfac5f6b71bb>

### Updated Production-Ready Specification: `public.event\_types\_master`

Version: 1.1

Date: May 18, 2025

#### 1\. Purpose & Primary Use-Cases

The `event\_types\_master` table defines the various categories or types of events that can occur (e.g., "Religious Feast Day," "Cultural Festival," "Local Market"). Its purpose is to provide a standardized, translatable classification for events, which helps in filtering, UI display (icons), and overall data consistency. Key user-story touchpoints include pilgrims filtering events by type, understanding event nature, admins categorizing events, and the system populating filters and icons.

#### 2\. Schema

| column | data\_type | constraints | description |

| `id` | `INTEGER` | `Primary Key, Generated always as identity` | Unique identifier for the event type. |

| `code` | `TEXT` | `Unique, Not Null, CHECK (length(code) > 0 AND length(code) <= 50 AND code ~ '^[a-z0-9\_]+$')` | Short, stable, machine-readable code (e.g., 'religious\_feast', 'cultural\_festival'). Snake\_case. |

| `default\_name` | `TEXT` | `Not Null, CHECK (length(default\_name) > 0 AND length(default\_name) <= 100)` | Human-readable name in the primary reference language. (Translatable via `public.translations`) |

| `default\_description` | `TEXT` | `Nullable` | Optional description in the primary reference language of the event type. (Translatable via `public.translations`) |

| `icon\_identifier` | `TEXT` | `Nullable, CHECK (icon\_identifier IS NULL OR length(icon\_identifier) <= 100)` | Name, class, or path for a UI icon associated with this event type. |

| `sort\_order` | `INTEGER` | `Not Null, Default 0` | Determines the display order in UI lists or filters. |

| `is\_active` | `BOOLEAN` | `Not Null, Default true` | If true, this event type is active and can be used. |

| `created\_at` | `TIMESTAMPTZ` | `Not Null, Default now()` | Timestamp of record creation. |

| `updated\_at` | `TIMESTAMPTZ` | `Not Null, Default now()` | Timestamp of last update (auto-updated by trigger). |

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

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

#### 3\. PostgreSQL DDL

SQL

```

CREATE TABLE public.event\_types\_master (

id INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

code TEXT UNIQUE NOT NULL CHECK (length(code) > 0 AND length(code) <= 50 AND code ~ '^[a-z0-9\_]+$'),

default\_name TEXT NOT NULL CHECK (length(default\_name) > 0 AND length(default\_name) <= 100),

default\_description TEXT NULL,

icon\_identifier TEXT NULL CHECK (icon\_identifier IS NULL OR length(icon\_identifier) <= 100),

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.event\_types\_master IS 'Master list of event types (e.g., festival, market, religious celebration). Replaces event\_type\_enum. Version 1.1';

COMMENT ON COLUMN public.event\_types\_master.id IS 'Unique identifier for the event type. Primary Key.';

COMMENT ON COLUMN public.event\_types\_master.code IS 'Short, stable, machine-readable code (snake\_case). Max 50 chars. Must be unique.';

COMMENT ON COLUMN public.event\_types\_master.default\_name IS 'Human-readable name in the primary reference language for UI. (Translatable via public.translations). Max 100 chars.';

COMMENT ON COLUMN public.event\_types\_master.default\_description IS 'Optional description in the primary reference language of the event type. (Translatable via public.translations).';

COMMENT ON COLUMN public.event\_types\_master.icon\_identifier IS 'Name, class, or path for a UI icon. Max 100 chars.';

COMMENT ON COLUMN public.event\_types\_master.sort\_order IS 'Determines the display order in UI lists or filters.';

COMMENT ON COLUMN public.event\_types\_master.is\_active IS 'If true, this event type is active and can be used.';

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

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

COMMENT ON COLUMN public.event\_types\_master.created\_by\_profile\_id IS 'Profile ID of the user who created this record. FK to public.profiles(id).';

COMMENT ON COLUMN public.event\_types\_master.updated\_by\_profile\_id IS 'Profile ID of the user who last updated this record. FK to public.profiles(id).';

-- Indexes

CREATE INDEX ix\_event\_types\_master\_active\_sort ON public.event\_types\_master (is\_active, sort\_order);

CREATE INDEX ix\_event\_types\_master\_created\_by ON public.event\_types\_master (created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX ix\_event\_types\_master\_updated\_by ON public.event\_types\_master (updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

```

#### 4\. Triggers/Functions

SQL

```

-- Standard updated\_at trigger (e.g., using extensions.moddatetime or a custom function)

CREATE TRIGGER handle\_event\_types\_master\_updated\_at

BEFORE UPDATE ON public.event\_types\_master

FOR EACH ROW

EXECUTE FUNCTION extensions.moddatetime('updated\_at'); -- Or public.set\_current\_timestamp\_updated\_at()

COMMENT ON TRIGGER handle\_event\_types\_master\_updated\_at ON public.event\_types\_master IS 'Trigger to automatically update updated\_at timestamp on row modification.';

-- Orphaned translation cleanup trigger

CREATE OR REPLACE FUNCTION public.cleanup\_event\_types\_master\_translations()

RETURNS TRIGGER AS $$

BEGIN

DELETE FROM public.translations

WHERE table\_identifier = 'event\_types\_master'

AND column\_identifier IN ('default\_name', 'default\_description') -- Specify translatable columns

AND row\_foreign\_key = OLD.id::TEXT;

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

COMMENT ON FUNCTION public.cleanup\_event\_types\_master\_translations() IS 'Cleans up orphaned translations from public.translations when an event\_types\_master record is deleted. Runs as SECURITY DEFINER. Ensure appropriate search\_path if not using fully qualified names.';

CREATE TRIGGER trigger\_cleanup\_event\_types\_master\_translations

AFTER DELETE ON public.event\_types\_master

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_event\_types\_master\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_event\_types\_master\_translations ON public.event\_types\_master IS 'After deleting an event type, remove its associated translations from the public.translations table.';

```

\*(Assumption: `extensions.moddatetime` is available or `public.set\_current\_timestamp\_updated\_at()` is defined as per project standards and `public.translations` table exists).\*

#### 5\. JSON Schema Mirror

JSON

```

{

"title": "event\_type\_master",

"description": "Master list of event types (e.g., festival, market, religious celebration). Replaces event\_type\_enum. Version 1.1.",

"type": "object",

"properties": {

"id": {

"type": "integer",

"description": "Unique identifier. Primary Key.",

"readOnly": true

},

"code": {

"type": "string",

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

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

"maxLength": 50

},

"default\_name": {

"type": "string",

"description": "Human-readable name in the primary reference language for UI. (Translatable via public.translations). Max 100 chars.",

"maxLength": 100

},

"default\_description": {

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

"description": "Optional description in the primary reference language of the event type. (Translatable via public.translations)."

},

"icon\_identifier": {

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

"maxLength": 100,

"description": "Name, class, or path for a UI icon."

},

"sort\_order": {

"type": "integer",

"default": 0,

"description": "Determines the display order in UI lists or filters."

},

"is\_active": {

"type": "boolean",

"default": true,

"description": "If true, this event type is active and can be used."

},

"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 (auto-updated by trigger).",

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who created this record. FK to public.profiles(id)."

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who last updated this record. FK to public.profiles(id)."

}

},

"required": [

"code",

"default\_name",

"sort\_order",

"is\_active"

],

"primary\_key": ["id"],

"unique\_constraints": [

{"columns": ["code"], "name": "event\_types\_master\_code\_key"}

],

"foreign\_keys": [

{"columns": ["created\_by\_profile\_id"], "references\_table": "public.profiles", "references\_columns": ["id"], "on\_delete": "SET NULL"},

{"columns": ["updated\_by\_profile\_id"], "references\_table": "public.profiles", "references\_columns": ["id"], "on\_delete": "SET NULL"}

]

}

```

#### 6\. Relationships & Integrity

- Primary Key: `id` (INTEGER)

- Unique Constraint: `code` must be unique.

- Foreign Key References FROM other tables:

- `events\_details.event\_type\_id` will reference `event\_types\_master.id` (ON DELETE RESTRICT recommended, to prevent deleting a type if it's in use).

- Array Foreign Key: None in this table.

#### 7\. Multilingual Strategy

- `default\_name`: Name in the primary reference language. (Translatable via `public.translations`)

- `default\_description`: Description in the primary reference language. (Translatable via `public.translations`)

- An `AFTER DELETE` trigger on this table (`trigger\_cleanup\_event\_types\_master\_translations`) ensures orphaned translations are removed from `public.translations`.

#### 8\. Role-Based Workflow & RLS Notes

- Key Fields: `is\_active` for lifecycle management.

- RLS Policies:

- 🟢 Allow public read access to all active event types.

SQL

```

CREATE POLICY "Allow public read access to active event types"

ON public.event\_types\_master FOR SELECT

USING (is\_active = true);

```

- 🟢 Allow authenticated users with 'admin\_platform' or 'admin\_super' roles to manage all event types.

SQL

```

CREATE POLICY "Allow platform admins to manage event types"

ON public.event\_types\_master FOR ALL

USING (

auth.role() = 'authenticated' AND

(SELECT public.has\_role\_on\_profile(auth.uid(), 'admin\_platform') OR public.has\_role\_on\_profile(auth.uid(), 'admin\_super'))

)

WITH CHECK (

auth.role() = 'authenticated' AND

(SELECT public.has\_role\_on\_profile(auth.uid(), 'admin\_platform') OR public.has\_role\_on\_profile(auth.uid(), 'admin\_super'))

);

```

- 🟠 `ON DELETE RESTRICT` from `events\_details` will prevent deletion if an event type is in use. Deactivation via `is\_active = false` is the preferred method for retiring types.

#### 9\. ENUM vs Lookup Discussion

- 🟢 Decision: This table `event\_types\_master` \*is\* the result of promoting the original `event\_type\_enum`.

- Reason: Essential for categorizing diverse events. Allows translatable names, descriptive text, icon association, and logical sorting for UI, providing a much richer and more maintainable system than a simple ENUM.

#### 10\. UI/UX Enablement

- `default\_name` (translated): For filter labels, event type display.

- `icon\_identifier`: Drives map icons for event waypoints or icons in event listings.

- `sort\_order`: Ensures logical presentation in filters.

- `default\_description` (translated): Can provide tooltips or further context about what an event type entails.

- `is\_active`: Used to filter types available in selection UIs for content managers.

#### 11\. Auditing & Lifecycle Management

- Audit Columns: `created\_at`, `updated\_at` (auto-managed by trigger `handle\_event\_types\_master\_updated\_at`). `created\_by\_profile\_id`, `updated\_by\_profile\_id` for admin change tracking.

- Lifecycle: `is\_active` flag manages whether the event type is available for use.

#### 12\. Scalability & Future-Proofing

- Lookup Table Structure: Easily supports adding new event types or modifying attributes.

- Manageable List: The number of distinct event types is expected to be moderate and stable.

#### 13\. Seed Data (V1.1)

| code | default\_name | default\_description | icon\_identifier | sort\_order | is\_active | created\_by\_profile\_id | updated\_by\_profile\_id |

| 'religious\_feast\_day\_procession' | 'Religious Feast/Procession' | 'Celebrations related to religious feast days, often including processions.' | 'icon-event-religious' | 10 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'local\_saint\_patron\_festival' | 'Local Saint/Patron Festival' | 'Festivals honoring a local patron saint.' | 'icon-event-saint' | 20 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'cultural\_historical\_festival' | 'Cultural/Historical Festival' | 'Events celebrating local culture, history, or traditions (e.g., medieval reenactments).' | 'icon-event-cultural' | 30 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'music\_arts\_festival' | 'Music/Arts Festival' | 'Festivals focused on music, visual arts, or performing arts.' | 'icon-event-music-arts' | 40 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'food\_wine\_sagra' | 'Food/Wine Festival (Sagra)' | 'Local festivals celebrating specific foods, wines, or agricultural products.' | 'icon-event-sagra' | 50 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'weekly\_market' | 'Weekly Market' | 'Regularly scheduled weekly markets for general goods or produce.' | 'icon-event-market-weekly' | 60 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'monthly\_antiques\_market' | 'Monthly Antiques/Craft Market' | 'Markets occurring monthly, often focused on antiques, crafts, or collectibles.' | 'icon-event-market-monthly' | 70 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'sporting\_event\_local' | 'Local Sporting Event' | 'Community sporting events or competitions.' | 'icon-event-sport' | 80 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'pilgrimage\_specific\_gathering' | 'Pilgrimage Specific Gathering' | 'Events or gatherings specifically for or by pilgrims.' | 'icon-event-pilgrimage' | 90 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'conference\_workshop' | 'Conference/Workshop' | 'Formal gatherings for specific topics, if relevant to pilgrims.' | 'icon-event-conference' | 100 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'public\_holiday\_observance' | 'Public Holiday Observance' | 'General public holiday observances that might impact services or attract crowds.' | 'icon-event-holiday' | 110 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'exhibition\_art\_craft' | 'Exhibition (Art/Craft)' | 'Temporary displays of art, crafts, or other items of interest.' | 'icon-event-exhibition' | 120 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'other\_event' | 'Other Event' | 'Miscellaneous events not fitting other categories.' | 'icon-event-other' | 900 | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

\*(Note: `[ADMIN\_UUID]` is a placeholder for an actual admin/system user profile ID used for seeding.)\*

#### 14\. Next-Action Checklist

- 🔴 Create Table: Execute the DDL for `public.event\_types\_master`.

- 🔴 Create Indexes: Execute DDL for `ix\_event\_types\_master\_active\_sort`, `ix\_event\_types\_master\_created\_by`, `ix\_event\_types\_master\_updated\_by`.

- 🔴 Implement `updated\_at` Trigger: Ensure `extensions.moddatetime` or equivalent `public.set\_current\_timestamp\_updated\_at()` is available and create `handle\_event\_types\_master\_updated\_at` trigger.

- 🔴 Implement Orphan Translation Cleanup Trigger: Create `public.cleanup\_event\_types\_master\_translations()` function and apply `trigger\_cleanup\_event\_types\_master\_translations` (AFTER `public.translations` table exists). Ensure `SECURITY DEFINER` functions have hardened `search\_path`.

- 🔴 RLS Policies: Review and apply defined RLS policies. Ensure `public.has\_role\_on\_profile` helper function exists.

- 🔴 Seed Data: Insert initial V1.1 data.

- 🟢 Iconography: Define `icon\_identifier` convention/source and prepare assets.

- 🟢 Population Logic: Define how `created\_by\_profile\_id` and `updated\_by\_profile\_id` will be populated by the application layer during admin operations.