<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\_recurrence\_frequencies\_master`

Version: 1.1

Date: May 18, 2025

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

The `event\_recurrence\_frequencies\_master` table defines the standard terms for how often a recurring event takes place (e.g., "Annually," "Monthly," "Weekly"). Its purpose is to provide a consistent, translatable vocabulary for describing event recurrence, aiding in user understanding and potential future logic for predicting event occurrences. Key user-story touchpoints include pilgrims understanding recurrence patterns, admins specifying frequencies, and the system displaying this information.

#### 2\. Schema

| column | data\_type | constraints | description |

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

| `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., 'annually', 'monthly'). 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 providing more context. (Translatable via `public.translations`) |

| `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 frequency 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\_recurrence\_frequencies\_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,

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\_recurrence\_frequencies\_master IS 'Master list of event recurrence frequencies (e.g., annually, monthly, weekly). Replaces event\_recurrence\_frequency\_enum. Version 1.1';

COMMENT ON COLUMN public.event\_recurrence\_frequencies\_master.id IS 'Unique identifier for the event recurrence frequency. Primary Key.';

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

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

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

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

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

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

COMMENT ON COLUMN public.event\_recurrence\_frequencies\_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\_recurrence\_frequencies\_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\_recurrence\_freq\_master\_active\_sort ON public.event\_recurrence\_frequencies\_master (is\_active, sort\_order);

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

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

```

#### 4\. Triggers/Functions

SQL

```

-- Standard updated\_at trigger

CREATE TRIGGER handle\_event\_recurrence\_freq\_master\_updated\_at

BEFORE UPDATE ON public.event\_recurrence\_frequencies\_master

FOR EACH ROW

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

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

-- Orphaned translation cleanup trigger

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

RETURNS TRIGGER AS $$

BEGIN

DELETE FROM public.translations

WHERE table\_identifier = 'event\_recurrence\_frequencies\_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\_recurrence\_freq\_master\_translations() IS 'Cleans up orphaned translations from public.translations when an event\_recurrence\_frequencies\_master record is deleted. Runs as SECURITY DEFINER. Ensure appropriate search\_path if not using fully qualified names.';

CREATE TRIGGER trigger\_cleanup\_event\_recurrence\_freq\_master\_translations

AFTER DELETE ON public.event\_recurrence\_frequencies\_master

FOR EACH ROW

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

COMMENT ON TRIGGER trigger\_cleanup\_event\_recurrence\_freq\_master\_translations ON public.event\_recurrence\_frequencies\_master IS 'After deleting a recurrence frequency, 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\_recurrence\_frequency\_master",

"description": "Master list of event recurrence frequencies (e.g., annually, monthly, weekly). Replaces event\_recurrence\_frequency\_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 frequency. (Translatable via public.translations)."

},

"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 frequency 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\_recurrence\_frequencies\_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.recurrence\_frequency\_id` will reference `event\_recurrence\_frequencies\_master.id`. Recommend `ON DELETE RESTRICT` for data integrity, ensuring a frequency definition is not removed if actively used.

- Array Foreign Key: None.

#### 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 (`trigger\_cleanup\_event\_recurrence\_freq\_master\_translations`) ensures orphaned translations are removed.

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

- Key Fields: `is\_active`.

- RLS Policies:

- 🟢 Allow public read access to active frequencies.

SQL

```

CREATE POLICY "Allow public read access to active event recurrence frequencies"

ON public.event\_recurrence\_frequencies\_master FOR SELECT

USING (is\_active = true);

```

- 🟢 Allow platform admins to manage frequencies.

SQL

```

CREATE POLICY "Allow platform admins to manage event recurrence frequencies"

ON public.event\_recurrence\_frequencies\_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` (when defined) will prevent deletion if a frequency is in use. Deactivation via `is\_active = false` is preferred.

#### 9\. ENUM vs Lookup Discussion

- 🟢 Decision: This table `event\_recurrence\_frequencies\_master` \*is\* the result of promoting the original `event\_recurrence\_frequency\_enum`.

- Reason: Provides translatable names, optional descriptive context, and a controlled vocabulary, offering better maintainability and clarity.

#### 10\. UI/UX Enablement

- `default\_name` (translated): For displaying the recurrence frequency (e.g., "Repeats: Annually").

- `default\_description` (translated): Can be used for tooltips or help text.

- `sort\_order`: To list frequencies logically (e.g., daily, weekly, monthly, annually).

- `is\_active`: Filters values for admin UIs.

#### 11\. Auditing & Lifecycle Management

- Audit Columns: `created\_at`, `updated\_at` (auto-managed). `created\_by\_profile\_id`, `updated\_by\_profile\_id` for admin tracking.

- Lifecycle: `is\_active` flag.

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

- Manageable List: The number of common recurrence frequencies is small and stable.

- Flexibility: Easy to add new frequencies or refine descriptions.

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

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

| 'daily' | 'Daily' | 10 | 'Event occurs every day.' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'weekly' | 'Weekly' | 20 | 'Event occurs once a week, typically on the same day(s).' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'bi\_weekly' | 'Bi-Weekly' | 30 | 'Event occurs every two weeks.' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'monthly' | 'Monthly' | 40 | 'Event occurs once a month, often on a specific day or week.' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'annually' | 'Annually' | 50 | 'Event occurs once a year, typically around the same date or period.' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

| 'other\_frequency' | 'Other Frequency' | 900 | 'Event recurs at a frequency not covered by standard options. See recurrence\_detail\_text for specifics.' | true | `[ADMIN\_UUID]` | `[ADMIN\_UUID]` |

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

#### 14\. Next-Action Checklist

- 🔴 Create Table: Execute DDL for `public.event\_recurrence\_frequencies\_master`.

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

- 🔴 Implement `updated\_at` Trigger: Ensure `extensions.moddatetime` or equivalent and create trigger.

- 🔴 Implement Orphan Translation Cleanup Trigger: Create `public.cleanup\_event\_recurrence\_freq\_master\_translations()` function and apply `trigger\_cleanup\_event\_recurrence\_freq\_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.

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

- 🟢 Review ON DELETE for FK from `events\_details` to this table when `events\_details` is finalized (recommend `ON DELETE RESTRICT`).