## [**https://gemini.google.com/u/1/app/1d4834db0819faf2**](https://gemini.google.com/u/1/app/1d4834db0819faf2)

https://gemini.google.com/u/1/app/d423d51e9f40da78

### 3\. Updated Production-Ready Specification (`attractions\_details`)

This document details the structure, purpose, and considerations for the `attractions\_details` table, Version 1.3.1 (reflecting RLS role name alignment). This version incorporates full V2 enhancements including robust audit fields, refined i18n strategy, array foreign key integrity with active status checks, media linking, and an `is\_active` check for its foreign key to `attraction\_types\_master`. It does \*not\* include denormalized English labels from master tables.

1\. Purpose & Primary Use-Cases

The `attractions\_details` table stores specific, rich information for waypoints that are attractions, cultural sites, historical landmarks, religious sites, or significant natural features. It extends the generic `waypoints` entry to include detailed descriptions, historical/cultural/spiritual significance, operational details like opening hours and entry fees, and available visitor amenities.

Key user-story touchpoints:

- Pilgrim: Discovering detailed information about an attraction (history, significance, what to see/do).

- Pilgrim: Finding practical details like opening hours, entry fees, tour availability, and accessibility.

- Pilgrim: Understanding the types of visitor amenities available on-site (e.g., toilets, cafe).

- Content Manager/Admin: Adding and curating comprehensive information about attractions, ensuring accuracy and depth.

- System/UI: Powering dedicated attraction pages with structured information, and enabling filtering by attraction type or features.

2\. Schema

| column | data\_type | constraints | description |

| `waypoint\_id` | `BIGINT` | Primary Key, Foreign Key to `public.waypoints(id)` ON DELETE CASCADE | Links to the generic `waypoints` table. This is the PK. Matches `waypoints.id` type. |

| `attraction\_type\_id` | `INTEGER` | Not Null, Foreign Key to `public.attraction\_types\_master(id)` ON DELETE RESTRICT | Specific type of attraction, linking to an \*active\* master table record. |

| `detailed\_description` | `TEXT` | Nullable | In-depth description, history, architecture, etc. Primary reference language (English) text. (Translatable via `public.translations`) |

| `historical\_significance\_notes` | `TEXT` | Nullable | Notes on historical importance. Primary reference language (English) text. (Translatable via `public.translations`) |

| `cultural\_significance\_notes` | `TEXT` | Nullable | Notes on cultural/artistic importance. Primary reference language (English) text. (Translatable via `public.translations`) |

| `spiritual\_significance\_notes` | `TEXT` | Nullable | Notes on spiritual or religious importance, especially for pilgrims. Primary reference language (English) text. (Translatable via `public.translations`) |

| `associated\_historical\_figures\_text` | `TEXT[]` | Nullable | Array of key historical/religious figures (free text). Elements are primary reference language (English) text. (Translatable via `public.translations` for each element) |

| `key\_historical\_events\_notes` | `TEXT` | Nullable | Notes on significant historical events that occurred here. Primary reference language (English) text. (Translatable via `public.translations`) |

| `opening\_hours\_structured` | `JSONB` | Nullable, CHECK (`opening\_hours\_structured` IS NULL OR jsonb\_type\_of(`opening\_hours\_structured`) = 'object') | Structured opening hours (e.g., based on schema.org OpeningHoursSpecification). Adheres to the centrally defined project schema. |

| `opening\_hours\_text\_notes` | `TEXT` | Nullable | Human-readable summary or additional notes about opening hours. Primary reference language (English) text. (Translatable via `public.translations`) |

| `opening\_hours\_last\_verified\_at` | `TIMESTAMPTZ` | Nullable | When the opening hours were last checked/verified. |

| `entry\_fee\_details` | `TEXT` | Nullable | Information on entry fees, donations, pilgrim discounts. Primary reference language (English) text. (Translatable via `public.translations`) |

| `guided\_tours\_info` | `TEXT` | Nullable | Information on availability, language, cost, and schedule of guided tours. Primary reference language (English) text. (Translatable via `public.translations`) |

| `audio\_guides\_info` | `TEXT` | Nullable | Information on availability, language, and cost of audio guides. Primary reference language (English) text. (Translatable via `public.translations`) |

| `photography\_allowed\_notes` | `TEXT` | Nullable | Policy on photography/videography. Primary reference language (English) text. (Translatable via `public.translations`) |

| `accessibility\_details\_specific` | `TEXT` | Nullable | Specific notes on accessibility for this attraction, supplementing `waypoints.waypoint\_accessibility\_notes`. Primary reference language (English) text. (Translatable via `public.translations`) |

| `visitor\_amenity\_ids` | `INTEGER[]` | Nullable | Array of Foreign Keys to `public.visitor\_amenities\_master(id)` for on-site amenities. Integrity (existence and active status) enforced by DB trigger. |

| `data\_last\_verified\_at` | `TIMESTAMPTZ` | Nullable | When the overall details for this attraction were last verified by an editor or reliable source. |

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

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

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

| `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 attraction record. |

3\. PostgreSQL DDL

SQL

```

-- Ensure prerequisite master tables are V2 compliant and created first:

-- public.waypoints (BIGINT id PK, content\_visibility\_status\_id INTEGER, deleted\_at TIMESTAMPTZ)

-- public.profiles (UUID id PK, roles TEXT[])

-- public.media (UUID id PK)

-- public.media\_roles\_master (TEXT code PK)

-- public.translations (for i18n)

-- public.languages\_master (for i18n)

-- public.content\_statuses\_master (INTEGER id PK, is\_publicly\_visible BOOLEAN)

-- Master table for Attraction Types (Version 1.1 - Assumed V2 Compliant)

CREATE TABLE IF NOT EXISTS public.attraction\_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\_]+$'),

label TEXT NOT NULL CHECK (length(label) > 0 AND length(label) <= 100), -- Primary reference language (English). Translatable.

description TEXT NULL, -- Primary reference language (English). Translatable.

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.attraction\_types\_master IS 'Master list of attraction types (e.g., church, museum). `label` and `description` are translatable. Version 1.1';

CREATE INDEX IF NOT EXISTS idx\_attraction\_types\_master\_is\_active ON public.attraction\_types\_master(is\_active);

CREATE TRIGGER trigger\_attraction\_types\_master\_set\_updated\_at

BEFORE UPDATE ON public.attraction\_types\_master FOR EACH ROW EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at();

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

RETURNS TRIGGER AS $$ BEGIN DELETE FROM public.translations WHERE table\_identifier = 'attraction\_types\_master' AND row\_foreign\_key = OLD.id::TEXT; RETURN OLD; END; $$ LANGUAGE plpgsql SECURITY DEFINER;

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

AFTER DELETE ON public.attraction\_types\_master FOR EACH ROW EXECUTE FUNCTION public.cleanup\_attraction\_types\_master\_translations();

-- Master table for Visitor Amenities (Version 1.1 - Assumed V2 Compliant)

CREATE TABLE IF NOT EXISTS public.visitor\_amenities\_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\_]+$'),

label TEXT NOT NULL CHECK (length(label) > 0 AND length(label) <= 100), -- Primary reference language (English). Translatable.

description TEXT NULL, -- Primary reference language (English). Translatable.

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

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.visitor\_amenities\_master IS 'Master list of on-site amenities for visitors at attractions (e.g., toilets, gift\_shop). `label` and `description` are translatable. Version 1.1';

CREATE INDEX IF NOT EXISTS idx\_visitor\_amenities\_master\_is\_active ON public.visitor\_amenities\_master(is\_active);

CREATE TRIGGER trigger\_visitor\_amenities\_master\_set\_updated\_at

BEFORE UPDATE ON public.visitor\_amenities\_master FOR EACH ROW EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at();

CREATE OR REPLACE FUNCTION public.cleanup\_visitor\_amenities\_master\_translations()

RETURNS TRIGGER AS $$ BEGIN DELETE FROM public.translations WHERE table\_identifier = 'visitor\_amenities\_master' AND row\_foreign\_key = OLD.id::TEXT; RETURN OLD; END; $$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_visitor\_amenities\_master\_translations

AFTER DELETE ON public.visitor\_amenities\_master FOR EACH ROW EXECUTE FUNCTION public.cleanup\_visitor\_amenities\_master\_translations();

-- Main Attractions Details Table (Version 1.3.1)

CREATE TABLE public.attractions\_details (

waypoint\_id BIGINT PRIMARY KEY,

attraction\_type\_id INTEGER NOT NULL,

-- No attraction\_type\_label\_en column

detailed\_description TEXT NULL,

historical\_significance\_notes TEXT NULL,

cultural\_significance\_notes TEXT NULL,

spiritual\_significance\_notes TEXT NULL,

associated\_historical\_figures\_text TEXT[] NULL,

key\_historical\_events\_notes TEXT NULL,

opening\_hours\_structured JSONB NULL CHECK (opening\_hours\_structured IS NULL OR jsonb\_type\_of(opening\_hours\_structured) = 'object'),

opening\_hours\_text\_notes TEXT NULL,

opening\_hours\_last\_verified\_at TIMESTAMPTZ NULL,

entry\_fee\_details TEXT NULL,

guided\_tours\_info TEXT NULL,

audio\_guides\_info TEXT NULL,

photography\_allowed\_notes TEXT NULL,

accessibility\_details\_specific TEXT NULL,

visitor\_amenity\_ids INTEGER[] NULL,

data\_last\_verified\_at TIMESTAMPTZ NULL,

created\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

created\_by\_profile\_id UUID NULL,

updated\_at TIMESTAMPTZ NOT NULL DEFAULT now(),

updated\_by\_profile\_id UUID NULL,

CONSTRAINT fk\_waypoint FOREIGN KEY(waypoint\_id) REFERENCES public.waypoints(id) ON DELETE CASCADE,

CONSTRAINT fk\_attraction\_type FOREIGN KEY(attraction\_type\_id) REFERENCES public.attraction\_types\_master(id) ON DELETE RESTRICT,

CONSTRAINT fk\_created\_by\_profile FOREIGN KEY(created\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL,

CONSTRAINT fk\_updated\_by\_profile FOREIGN KEY(updated\_by\_profile\_id) REFERENCES public.profiles(id) ON DELETE SET NULL

);

COMMENT ON TABLE public.attractions\_details IS 'Specific details for attractions, cultural sites, historical landmarks, etc., extending the waypoints table. Version 1.3.1';

COMMENT ON COLUMN public.attractions\_details.waypoint\_id IS 'Links to the generic waypoints table. This is the PK.';

COMMENT ON COLUMN public.attractions\_details.attraction\_type\_id IS 'FK to attraction\_types\_master. Specific type of attraction. Referenced master record must be is\_active=true.';

COMMENT ON COLUMN public.attractions\_details.detailed\_description IS 'In-depth description, history, architecture. Primary reference language (English) text. (Translatable via public.translations)';

COMMENT ON COLUMN public.attractions\_details.visitor\_amenity\_ids IS 'Array of FKs to visitor\_amenities\_master. On-site amenities. Integrity (existence and active status) of array elements MUST be enforced by a dedicated database trigger.';

-- (Other column comments remain as previously specified for V1.3)

-- Linking table for Attraction Media (public.attraction\_details\_media - Version 1.0)

CREATE TABLE IF NOT EXISTS public.attraction\_details\_media (

id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

attraction\_waypoint\_id BIGINT NOT NULL REFERENCES public.attractions\_details(waypoint\_id) ON DELETE CASCADE,

media\_id UUID NOT NULL REFERENCES public.media(id) ON DELETE RESTRICT,

media\_role\_code TEXT NOT NULL REFERENCES public.media\_roles\_master(code) ON DELETE RESTRICT,

display\_order SMALLINT NOT NULL DEFAULT 0,

caption\_override TEXT NULL, -- Primary reference language (English). Translatable.

alt\_text\_override TEXT NULL, -- Primary reference language (English). Translatable.

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,

CONSTRAINT uq\_attraction\_media\_role\_order UNIQUE (attraction\_waypoint\_id, media\_id, media\_role\_code)

);

COMMENT ON TABLE public.attraction\_details\_media IS 'Links attractions to media items for galleries or specific semantic roles. Version 1.0';

CREATE TRIGGER trigger\_attraction\_details\_media\_set\_updated\_at

BEFORE UPDATE ON public.attraction\_details\_media FOR EACH ROW EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at();

CREATE OR REPLACE FUNCTION public.cleanup\_attraction\_details\_media\_translations()

RETURNS TRIGGER AS $$ BEGIN DELETE FROM public.translations WHERE table\_identifier = 'attraction\_details\_media' AND row\_foreign\_key = OLD.id::TEXT; RETURN OLD; END; $$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_attraction\_details\_media\_translations

AFTER DELETE ON public.attraction\_details\_media FOR EACH ROW EXECUTE FUNCTION public.cleanup\_attraction\_details\_media\_translations();

-- Triggers & Functions for attractions\_details

CREATE TRIGGER trigger\_attractions\_details\_set\_updated\_at

BEFORE UPDATE ON public.attractions\_details

FOR EACH ROW EXECUTE FUNCTION public.set\_current\_timestamp\_updated\_at();

COMMENT ON TRIGGER trigger\_attractions\_details\_set\_updated\_at ON public.attractions\_details IS 'Trigger to automatically update updated\_at timestamp on row modification.';

CREATE OR REPLACE FUNCTION public.check\_attraction\_visitor\_amenities()

RETURNS TRIGGER AS $$ DECLARE amenity\_id INTEGER; is\_valid BOOLEAN; amenity\_is\_active BOOLEAN; BEGIN IF NEW.visitor\_amenity\_ids IS NOT NULL AND array\_length(NEW.visitor\_amenity\_ids, 1) > 0 THEN FOREACH amenity\_id IN ARRAY NEW.visitor\_amenity\_ids LOOP SELECT EXISTS (SELECT 1 FROM public.visitor\_amenities\_master WHERE id = amenity\_id), v.is\_active INTO is\_valid, amenity\_is\_active FROM public.visitor\_amenities\_master v WHERE v.id = amenity\_id; IF NOT is\_valid THEN RAISE EXCEPTION 'Invalid visitor\_amenity\_id: % does not exist in public.visitor\_amenities\_master.', amenity\_id; END IF; IF NOT amenity\_is\_active THEN RAISE EXCEPTION 'Inactive visitor\_amenity\_id: % is not active in public.visitor\_amenities\_master.', amenity\_id; END IF; END LOOP; END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_check\_attraction\_visitor\_amenities

BEFORE INSERT OR UPDATE OF visitor\_amenity\_ids ON public.attractions\_details

FOR EACH ROW EXECUTE FUNCTION public.check\_attraction\_visitor\_amenities();

COMMENT ON TRIGGER trigger\_check\_attraction\_visitor\_amenities ON public.attractions\_details IS 'Ensures all visitor\_amenity\_ids exist in visitor\_amenities\_master and are active.';

CREATE OR REPLACE FUNCTION public.cleanup\_attraction\_details\_translations()

RETURNS TRIGGER AS $$ BEGIN DELETE FROM public.translations WHERE table\_identifier = 'attractions\_details' AND row\_foreign\_key = OLD.waypoint\_id::TEXT; DELETE FROM public.translations WHERE table\_identifier = 'attraction\_details\_media' AND row\_foreign\_key IN (SELECT adm.id::TEXT FROM public.attraction\_details\_media adm WHERE adm.attraction\_waypoint\_id = OLD.waypoint\_id); RETURN OLD; END; $$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_attraction\_details\_translations

AFTER DELETE ON public.attractions\_details

FOR EACH ROW EXECUTE FUNCTION public.cleanup\_attraction\_details\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_attraction\_details\_translations ON public.attractions\_details IS 'Cleans up orphaned translations from public.translations when an attraction\_details record (and its media links) are deleted.';

CREATE OR REPLACE FUNCTION public.check\_attraction\_type\_active()

RETURNS TRIGGER AS $$ DECLARE is\_type\_active BOOLEAN; BEGIN IF NEW.attraction\_type\_id IS NOT NULL THEN SELECT atm.is\_active INTO is\_type\_active FROM public.attraction\_types\_master atm WHERE atm.id = NEW.attraction\_type\_id; IF NOT is\_type\_active THEN RAISE EXCEPTION 'Cannot link attraction detail to inactive attraction\_type\_id: %. Corresponding attraction\_types\_master record must be active.', NEW.attraction\_type\_id; END IF; END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_check\_attraction\_type\_active

BEFORE INSERT OR UPDATE OF attraction\_type\_id ON public.attractions\_details

FOR EACH ROW EXECUTE FUNCTION public.check\_attraction\_type\_active();

COMMENT ON TRIGGER trigger\_check\_attraction\_type\_active ON public.attractions\_details IS 'Ensures the referenced attraction\_type\_id in attraction\_types\_master is active.';

-- Indexes

CREATE INDEX idx\_attractions\_details\_attraction\_type\_id ON public.attractions\_details(attraction\_type\_id);

CREATE INDEX idx\_attractions\_details\_visitor\_amenity\_ids ON public.attractions\_details USING GIN (visitor\_amenity\_ids) WHERE visitor\_amenity\_ids IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_attractions\_details\_opening\_hours\_structured ON public.attractions\_details USING GIN (opening\_hours\_structured) WHERE opening\_hours\_structured IS NOT NULL;

CREATE INDEX idx\_adm\_created\_by\_profile\_id ON public.attractions\_details(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX idx\_adm\_updated\_by\_profile\_id ON public.attractions\_details(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

CREATE INDEX idx\_attraction\_details\_media\_attraction\_id ON public.attraction\_details\_media(attraction\_waypoint\_id);

CREATE INDEX idx\_attraction\_details\_media\_media\_id ON public.attraction\_details\_media(media\_id);

CREATE INDEX idx\_attraction\_details\_media\_role\_code ON public.attraction\_details\_media(media\_role\_code);

-- RLS Policies

ALTER TABLE public.attractions\_details ENABLE ROW LEVEL SECURITY;

CREATE POLICY "public\_read\_attractions\_details" ON public.attractions\_details

FOR SELECT USING (

EXISTS (

SELECT 1 FROM public.waypoints w

JOIN public.content\_statuses\_master csm ON w.content\_visibility\_status\_id = csm.id

WHERE w.id = attractions\_details.waypoint\_id

AND csm.is\_publicly\_visible = TRUE

AND w.deleted\_at IS NULL

)

);

CREATE POLICY "manage\_attractions\_details\_for\_privileged\_users" ON public.attractions\_details

FOR ALL

USING (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'platform\_admin')) OR -- Updated role name

(SELECT public.has\_role\_on\_profile(auth.uid(), 'regional\_content\_manager')) -- Updated role name

)

WITH CHECK (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'platform\_admin')) OR

(SELECT public.has\_role\_on\_profile(auth.uid(), 'regional\_content\_manager'))

AND (TG\_OP = 'INSERT' OR (TG\_OP = 'UPDATE' AND OLD.waypoint\_id = NEW.waypoint\_id OR (SELECT public.has\_role\_on\_profile(auth.uid(), 'platform\_admin'))))

);

ALTER TABLE public.attraction\_details\_media ENABLE ROW LEVEL SECURITY;

CREATE POLICY "public\_read\_attraction\_media" ON public.attraction\_details\_media

FOR SELECT USING (

EXISTS (

SELECT 1 FROM public.attractions\_details ad

JOIN public.waypoints w ON ad.waypoint\_id = w.id

JOIN public.content\_statuses\_master csm ON w.content\_visibility\_status\_id = csm.id

WHERE ad.waypoint\_id = attraction\_details\_media.attraction\_waypoint\_id

AND csm.is\_publicly\_visible = TRUE

AND w.deleted\_at IS NULL

)

);

CREATE POLICY "manage\_attraction\_media\_for\_privileged\_users" ON public.attraction\_details\_media

FOR ALL

USING (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'platform\_admin')) OR -- Updated role name

(SELECT public.has\_role\_on\_profile(auth.uid(), 'regional\_content\_manager')) -- Updated role name

)

WITH CHECK (

(SELECT public.has\_role\_on\_profile(auth.uid(), 'platform\_admin')) OR

(SELECT public.has\_role\_on\_profile(auth.uid(), 'regional\_content\_manager'))

);

```

4\. JSON Schema Mirror

(Reflects the schema table in Section 2; does not include attraction\_type\_label\_en)

JSON

```

{

"title": "attraction\_detail",

"description": "Specific details for attractions, cultural sites, historical landmarks, etc., extending the waypoints table. Version 1.3.1",

"type": "object",

"properties": {

"waypoint\_id": {

"type": "integer",

"format": "int64",

"description": "Links to the generic waypoints table. This is the PK."

},

"attraction\_type\_id": {

"type": "integer",

"description": "FK to attraction\_types\_master. Specific type of attraction. Referenced master record must be is\_active=true."

},

"detailed\_description": {

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

"description": "In-depth description, history, architecture. Primary reference language (English) text. (Translatable via public.translations)"

},

"historical\_significance\_notes": {

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

"description": "Notes on historical importance. Primary reference language (English) text. (Translatable via public.translations)"

},

"cultural\_significance\_notes": {

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

"description": "Notes on cultural/artistic importance. Primary reference language (English) text. (Translatable via public.translations)"

},

"spiritual\_significance\_notes": {

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

"description": "Notes on spiritual or religious importance for pilgrims. Primary reference language (English) text. (Translatable via public.translations)"

},

"associated\_historical\_figures\_text": {

"type": ["array", "null"],

"items": { "type": "string" },

"description": "Array of key historical/religious figures (free text). Elements are primary reference language (English) text. (Translatable via public.translations for each element)"

},

"key\_historical\_events\_notes": {

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

"description": "Notes on significant historical events. Primary reference language (English) text. (Translatable via public.translations)"

},

"opening\_hours\_structured": {

"type": ["object", "null"],

"description": "Structured opening hours in JSONB format. Adheres to the centrally defined schema (e.g., based on schema.org/OpeningHoursSpecification).",

"additionalProperties": true

},

"opening\_hours\_text\_notes": {

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

"description": "Human-readable summary or notes about opening hours. Primary reference language (English) text. (Translatable via public.translations)"

},

"opening\_hours\_last\_verified\_at": {

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

"format": "date-time",

"description": "When the opening hours were last checked/verified."

},

"entry\_fee\_details": {

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

"description": "Information on entry fees, donations, pilgrim discounts. Primary reference language (English) text. (Translatable via public.translations)"

},

"guided\_tours\_info": {

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

"description": "Information on guided tours. Primary reference language (English) text. (Translatable via public.translations)"

},

"audio\_guides\_info": {

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

"description": "Information on audio guides. Primary reference language (English) text. (Translatable via public.translations)"

},

"photography\_allowed\_notes": {

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

"description": "Policy on photography/videography. Primary reference language (English) text. (Translatable via public.translations)"

},

"accessibility\_details\_specific": {

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

"description": "Specific accessibility notes for this attraction. Primary reference language (English) text. (Translatable via public.translations)"

},

"visitor\_amenity\_ids": {

"type": ["array", "null"],

"items": { "type": "integer" },

"description": "Array of FKs to visitor\_amenities\_master. On-site amenities. Integrity (existence and active status) enforced by DB trigger."

},

"data\_last\_verified\_at": {

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

"format": "date-time",

"description": "When the overall details for this attraction were last verified."

},

"created\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of record creation.",

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who created this attraction record."

},

"updated\_at": {

"type": "string",

"format": "date-time",

"description": "Timestamp of last update.",

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who last updated this attraction record."

}

},

"required": [

"waypoint\_id",

"attraction\_type\_id",

"created\_at",

"updated\_at"

]

}

```

5\. Relationships & Integrity

- Primary Key: `waypoint\_id` (`BIGINT`)

- Foreign Keys:

- `waypoint\_id` REFERENCES `public.waypoints(id)` ON DELETE CASCADE

- `attraction\_type\_id` REFERENCES `public.attraction\_types\_master(id)` ON DELETE RESTRICT

- Integrity constraint: Referenced `attraction\_types\_master` record must have `is\_active = true` (enforced by `trigger\_check\_attraction\_type\_active`).

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

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

- Array Foreign Key: `visitor\_amenity\_ids` (elements are `INTEGER`) conceptually references `public.visitor\_amenities\_master(id)`.

- Integrity constraint: Each ID must exist in `visitor\_amenities\_master` and the referenced record must have `is\_active = true` (enforced by `trigger\_check\_attraction\_visitor\_amenities`).

- Linking Tables:

- `public.attraction\_details\_media` (Version 1.0) links this table to `public.media` for galleries or specific media roles. Its own DDL includes audit columns, `updated\_at` trigger, and orphan translation cleanup for its translatable fields.

- Child Table: This table is the parent for `public.religious\_service\_schedules` (which links via `attractions\_details.waypoint\_id`).

6\. Multilingual Strategy

- Directly Translatable Fields (Primary reference language: English, translated via `public.translations`):

- `detailed\_description`, `historical\_significance\_notes`, `cultural\_significance\_notes`, `spiritual\_significance\_notes`, elements within `associated\_historical\_figures\_text`, `key\_historical\_events\_notes`, `opening\_hours\_text\_notes`, `entry\_fee\_details`, `guided\_tours\_info`, `audio\_guides\_info`, `photography\_allowed\_notes`, `accessibility\_details\_specific`.

- Translatable fields from `attraction\_details\_media`: `caption\_override`, `alt\_text\_override`.

- Indirectly Translatable Fields (via `label` fields in master tables, fetched by views/application logic):

- `attraction\_type\_id` (links to `attraction\_types\_master.label`)

- `visitor\_amenity\_ids` (elements link to `visitor\_amenities\_master.label`)

- Orphan Cleanup: `trigger\_cleanup\_attraction\_details\_translations` `AFTER DELETE` on `attractions\_details` removes related entries from `public.translations` for `attractions\_details` fields and for fields from linked `attraction\_details\_media` records. Master tables (`attraction\_types\_master`, `visitor\_amenities\_master`) and `attraction\_details\_media` itself have their own triggers for cleaning their respective translations.

7\. Role-Based Workflow & RLS Notes

- Key Fields for Workflow: `data\_last\_verified\_at`, `opening\_hours\_last\_verified\_at` for content freshness. Audit columns for tracking. Publication status is driven by the parent `waypoints` record.

- RLS Policies:

- Row-Level Security is enabled on `attractions\_details` and `attraction\_details\_media`.

- Public users: `SELECT` access to details of published and non-deleted attractions (based on parent `waypoints` status and `content\_statuses\_master.is\_publicly\_visible`).

- Privileged users (e.g., `platform\_admin`, `regional\_content\_manager` via `public.has\_role\_on\_profile`): `ALL` operations, with checks to prevent `waypoint\_id` reparenting by non-`platform\_admin` roles.

8\. ENUM vs Lookup Discussion

- `attraction\_types\_master`: Promoted from an ENUM. Allows richer attributes (translatable labels/descriptions, icons, sort order, `is\_active` flag, audit).

- `visitor\_amenities\_master`: `visitor\_amenity\_ids` links to this master table, enabling standardized, translatable amenities with icons and `is\_active` flags.

9\. UI/UX Enablement

- Filters, Icons, Lists: Driven by `attraction\_type\_id` (via `attraction\_types\_master`) and `visitor\_amenity\_ids` (via `visitor\_amenities\_master`).

- Dynamic Content: `opening\_hours\_structured` for "Open Now?" indicators.

- Rich Information: `entry\_fee\_details`, `detailed\_description`, etc., for attraction pages.

- Media Display: `attraction\_details\_media` table supports image galleries.

- Performance: Key indexes on FKs, GIN indexes on array/JSONB fields.

10\. Key Considerations & Definitions

- Data Integrity:

- `attractions\_details` requires a parent `waypoints` record.

- FKs to master tables (`attraction\_types\_master`, `visitor\_amenities\_master`) are further protected by triggers ensuring master records are `is\_active = true`.

- `opening\_hours\_structured` (JSONB): Adherence to a central JSON schema is crucial.

- `associated\_historical\_figures\_text` (TEXT[]): Elements are translatable free text. Future versions might use a dedicated master table.

- Data Freshness: `data\_last\_verified\_at` and `opening\_hours\_last\_verified\_at` are important.

11\. Scalability & Future-Proofing

- Lookup tables, `JSONB`, audit columns, and media linking provide good scalability. Partitioning would follow `waypoints`.

12\. Next-Action Checklist

- 🔴 Verify Prerequisite Master Tables: Confirm `public.attraction\_types\_master` (V1.1) and `public.visitor\_amenities\_master` (V1.1) are implemented with `is\_active`, `label`, full audit, and relevant triggers. Seed data populated.

- 🔴 Verify Global Prerequisites: `public.waypoints`, `public.profiles`, `public.media`, `public.media\_roles\_master`, `public.translations`, `public.languages\_master`, `public.content\_statuses\_master` are V2 compliant.

- 🔴 Implement/Verify `attractions\_details` Table: Execute DDL for Version 1.3.1 (ensuring no `\_en` denormalized columns).

- 🔴 Implement/Verify Triggers on `attractions\_details`:

- `trigger\_attractions\_details\_set\_updated\_at`.

- `trigger\_check\_attraction\_visitor\_amenities` (checks existence and `is\_active`).

- `trigger\_cleanup\_attraction\_details\_translations` (handles main table and linked media).

- `trigger\_check\_attraction\_type\_active`.

- 🔴 Implement/Verify `attraction\_details\_media` Table: DDL (Version 1.0), triggers (`updated\_at`, orphan translation for its fields), indexes, RLS are correct.

- 🟠 JSON Schema for `opening\_hours\_structured`: Formalize and document.

- 🟠 RLS Helper Functions: Ensure `public.has\_role\_on\_profile(UUID, TEXT)` is robustly defined and tested.

- 🟢 RLS Policies: Thoroughly test all implemented RLS policies.

- 🟢 Translation Entries: Prepare for all translatable fields.

- 🟢 Data Migration: If prior data exists, ensure schema compatibility.