New View Spec: v\_amenities\_localized

### **1. View Name**

public.v\_amenities\_localized

### **2. Purpose & Primary Use-Cases**

* **Purpose**: To provide a consolidated and denormalized view of amenities\_master records, including all their available translations for name, description, and amenity\_category fields in a structured JSONB format, alongside all other master table attributes.
* **Primary Use-Cases**:
  + Simplify API development by providing a single source for an amenity and all its translations.
  + Support API endpoints that list amenities or display details for a specific amenity where multilingual representation is required (e.g., for filtering accommodations or displaying available amenities).
  + Facilitate easier data fetching for frontend components that need to display localized amenity information.

### **3. View Schema (Columns)**

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | integer | Inherited from amenities\_master. Unique identifier for the amenity. |
| amenity\_code | text | Inherited from amenities\_master. Short, stable, machine-readable code (e.g., 'WIFI\_INTERNET'). |
| name | text | Inherited from amenities\_master. Primary reference language (English) name for the amenity. |
| description | text | Inherited from amenities\_master. Optional primary reference language (English) description of the amenity. |
| icon\_identifier | text | Inherited from amenities\_master. Name, class, or path for a UI icon. |
| amenity\_category | text | Inherited from amenities\_master. Broad category for grouping amenities in the primary reference language (English) (e.g., "Connectivity", "Services"). |
| is\_common\_pilgrim\_need | boolean | Inherited from amenities\_master. Flags amenities particularly sought after by pilgrims. |
| sort\_order | integer | Inherited from amenities\_master. Determines display order in UI lists. |
| is\_active | boolean | Inherited from amenities\_master. True if the amenity is active and available for use. |
| created\_at | timestamp with time zone | Inherited from amenities\_master. Timestamp of record creation. |
| updated\_at | timestamp with time zone | Inherited from amenities\_master. Timestamp of last update. |
| created\_by\_profile\_id | uuid | Inherited from amenities\_master. Profile ID of the user/admin who created this record. |
| updated\_by\_profile\_id | uuid | Inherited from amenities\_master. Profile ID of the user/admin who last updated this record. |
| all\_translations | jsonb | A JSONB object containing all available translations for name, description, and amenity\_category, keyed by language code. Example: {"en": {"name": "...", "description": "...", "amenity\_category": "..."}, "it": {"name": "...", "description": "...", "amenity\_category": "..."}}. |

### **4. Underlying SQL Definition**

SQL

CREATE OR REPLACE VIEW public.v\_amenities\_localized AS

SELECT

am.id,

am.amenity\_code,

am.name, -- Primary English name

am.description, -- Primary English description

am.icon\_identifier,

am.amenity\_category, -- Primary English amenity\_category

am.is\_common\_pilgrim\_need,

am.sort\_order,

am.is\_active,

am.created\_at,

am.updated\_at,

am.created\_by\_profile\_id,

am.updated\_by\_profile\_id,

(

SELECT

jsonb\_object\_agg(

tr.language\_code,

jsonb\_build\_object(

'name', MAX(CASE WHEN tr.column\_identifier = 'name' THEN tr.translated\_text ELSE NULL END),

'description', MAX(CASE WHEN tr.column\_identifier = 'description' THEN tr.translated\_text ELSE NULL END),

'amenity\_category', MAX(CASE WHEN tr.column\_identifier = 'amenity\_category' THEN tr.translated\_text ELSE NULL END)

)

)

FROM public.translations tr

WHERE

tr.table\_identifier = 'amenities\_master'

AND tr.row\_foreign\_key = am.id::TEXT

GROUP BY tr.row\_foreign\_key

) AS all\_translations

FROM

public.amenities\_master am;

COMMENT ON VIEW public.v\_amenities\_localized IS 'Provides amenities with their base English fields and a JSONB column "all\_translations" containing all available name, description, and amenity\_category translations keyed by language code. Version 1.0';

COMMENT ON COLUMN public.v\_amenities\_localized.name IS 'Primary reference language (English) name from amenities\_master.';

COMMENT ON COLUMN public.v\_amenities\_localized.description IS 'Primary reference language (English) description from amenities\_master.';

COMMENT ON COLUMN public.v\_amenities\_localized.amenity\_category IS 'Primary reference language (English) amenity\_category from amenities\_master.';

COMMENT ON COLUMN public.v\_amenities\_localized.all\_translations IS 'JSONB object with all translations for name, description, and amenity\_category, keyed by language code. E.g., {"en": {"name": "...", "description": "...", "amenity\_category": "..."}, "it": {"name": "...", "description": "...", "amenity\_category": "..."}}. Base English text from master table should be merged by application/API layer if not also present in translations table with code ''en''.';

### **5. Key Dependencies**

* public.amenities\_master (Version 1.4 or later, which includes is\_active and audit columns)
* public.translations (Version 2.1 or later)

### **6. Performance Considerations**

* The subquery using jsonb\_object\_agg can be resource-intensive without proper indexing on public.translations.
* **Required Index on** public.translations: A composite index on (table\_identifier, row\_foreign\_key, language\_code, column\_identifier) is crucial. (Assumed to be idx\_translations\_lookup\_multi\_col).
* Queries on this view filtering by is\_active will benefit from the idx\_am\_is\_active index on amenities\_master.
* For very high-read scenarios, a materialized view could be a V2+ optimization.

### **7. RLS & Security Notes**

* RLS policies from public.amenities\_master (e.g., filtering by is\_active = true) will be inherited by this view.
* Access to public.translations is assumed if a user can see the master record.
* Define with SECURITY INVOKER (default for views).

### **8. API Endpoints Supported (Conceptual)**

* GET /meta/amenities?lang=it: List all active amenities with translations.
* GET /meta/amenities/{amenity\_code}: Get a specific amenity with translations.
* Used by accommodation-related endpoints to list available amenities with localized details.

### **9. Rationale for Creation**

* Decouples API data retrieval from complex translation join/aggregation logic for multiple fields.
* Provides a consistent, denormalized structure for accessing localized amenity content.
* Improves query readability and maintainability for developers.

### **10. Key Considerations & Definitions**

* **Primary Language Fallback**: The API/application layer needs to handle fallback to primary English fields if a translation for a requested language is not available in all\_translations.
* **Data Freshness**: As a standard view, it reflects the current state.

### **11. Scalability & Future-Proofing**

* Performance depends on translations table indexing.
* Adding new translatable fields to amenities\_master would require updating this view definition.

### **12. Next-Action Checklist**

* 🔴 **Create View**: Execute the DDL to create public.v\_amenities\_localized.
* 🔴 **Verify Index on** translations: Ensure the recommended composite index on public.translations exists.
* 🟠 **API Layer Integration**: Plan API consumption of this view.
* 🟢 **Testing**: Test view performance.
* 🟢 **Documentation**: Document this view.