https://gemini.google.com/u/1/app/21b8e85b3e611787

New View Spec: v\_accommodation\_types\_localized

### **1. View Name**

public.v\_accommodation\_types\_localized

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

* **Purpose**: To provide a consolidated and denormalized view of accommodation\_types\_master records, including all their available translations for label and description 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 accommodation type and all its translations, reducing complex joins in the API layer.
  + Support API endpoints that list accommodation types or display details for a specific accommodation type where multilingual representation is required.
  + Facilitate easier data fetching for frontend components that need to display localized accommodation type information (e.g., filter dropdowns, category displays).

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

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | integer | Inherited from accommodation\_types\_master. Unique identifier for the accommodation type. |
| code | text | Inherited from accommodation\_types\_master. Short, stable, machine-readable code (e.g., 'ospitale\_pilgrim\_hostel'). |
| label | text | Inherited from accommodation\_types\_master. Primary reference language (English) label for the accommodation type. |
| description | text | Inherited from accommodation\_types\_master. Optional primary reference language (English) description of the accommodation type. |
| icon\_identifier | text | Inherited from accommodation\_types\_master. Name, class, or path for a UI icon. |
| sort\_order | integer | Inherited from accommodation\_types\_master. Determines display order in UI lists. |
| is\_active | boolean | Inherited from accommodation\_types\_master. True if the accommodation type is active and available for use. |
| created\_at | timestamp with time zone | Inherited from accommodation\_types\_master. Timestamp of record creation. |
| updated\_at | timestamp with time zone | Inherited from accommodation\_types\_master. Timestamp of last update. |
| created\_by\_profile\_id | uuid | Inherited from accommodation\_types\_master. Profile ID of the user/admin who created this record. |
| updated\_by\_profile\_id | uuid | Inherited from accommodation\_types\_master. Profile ID of the user/admin who last updated this record. |
| all\_translations | jsonb | A JSONB object containing all available translations for label and description, keyed by language code. Example: {"en": {"label": "...", "description": "..."}, "it": {"label": "...", "description": "..."}}. Includes primary English text if present in translations table. |

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

SQL

CREATE OR REPLACE VIEW public.v\_accommodation\_types\_localized AS

SELECT

atm.id,

atm.code,

atm.label, -- Primary English label

atm.description, -- Primary English description

atm.icon\_identifier,

atm.sort\_order,

atm.is\_active,

atm.created\_at,

atm.updated\_at,

atm.created\_by\_profile\_id,

atm.updated\_by\_profile\_id,

(

SELECT

jsonb\_object\_agg(

tr.language\_code,

jsonb\_build\_object(

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

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

)

)

FROM public.translations tr

WHERE

tr.table\_identifier = 'accommodation\_types\_master'

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

GROUP BY tr.row\_foreign\_key -- Though PK is unique, GROUP BY is needed for agg

) AS all\_translations

FROM

public.accommodation\_types\_master atm;

COMMENT ON VIEW public.v\_accommodation\_types\_localized IS 'Provides accommodation types with their base English fields and a JSONB column "all\_translations" containing all available label and description translations keyed by language code. Version 1.0';

COMMENT ON COLUMN public.v\_accommodation\_types\_localized.label IS 'Primary reference language (English) label from accommodation\_types\_master.';

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

COMMENT ON COLUMN public.v\_accommodation\_types\_localized.all\_translations IS 'JSONB object with all translations for label and description, keyed by language code. E.g., {"en": {"label": "...", "description": "..."}, "it": {"label": "...", "description": "..."}}. 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.accommodation\_types\_master (Version 1.3 or later, which includes is\_active and audit columns)
* public.translations (Version 2.1 or later, with table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translated\_text)

### **6. Performance Considerations**

* The subquery using jsonb\_object\_agg to gather translations can be resource-intensive if the translations table is very large and not adequately indexed.
* **Required Index on** public.translations: A composite index on (table\_identifier, row\_foreign\_key, language\_code, column\_identifier) is crucial for the performance of the subquery.
* SQL

CREATE INDEX IF NOT EXISTS idx\_translations\_lookup\_multi\_col ON public.translations(table\_identifier, row\_foreign\_key, language\_code, column\_identifier);

* Queries on this view filtering by is\_active from the base table will benefit from the idx\_atm\_is\_active index on accommodation\_types\_master.
* The view is not materialized. For very high-read scenarios with slowly changing master data and translations, a materialized view could be a V2+ optimization.

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

* Row-Level Security policies applied to the underlying public.accommodation\_types\_master table (e.g., filtering by is\_active = true for public roles) will be inherited and automatically applied when this view is queried by those roles.
* Access to public.translations should generally be permissible if a user has access to the master record, but RLS on translations (if any) could also affect results. Typically, translations are considered part of the content they translate.
* The view itself does not require separate RLS policies if the base table's RLS is sufficient. Define with SECURITY INVOKER (default for views).

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

* Primarily supports GET requests for listing all active accommodation types or fetching a specific type by ID/code, where translations are needed in a nested structure.
  + Example: GET /meta/accommodation-types?lang=it (API layer would use all\_translations to provide the Italian version).
  + Example: GET /meta/accommodation-types/{type\_code} (could return the primary language fields directly and the all\_translations object).
* Reduces the need for the API to perform complex translation joins for each request.

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

* **Simplifies API Development**: Abstracts the complexity of joining master data with translations.
* **Consistent Data Structure**: Provides a standardized way to access localized content for accommodation types.
* **Improved Query Readability**: Developers querying the view have a simpler interface.
* **Potential Performance Benefit**: While the view itself does joins, it centralizes this logic. If materialized (V2+), it could offer significant read performance gains.

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

* **Primary Language Fallback**: The view provides the primary English label and description directly from the master table. The all\_translations JSONB object contains translations from the translations table. The API or application layer will need to handle the logic of using a specific language from all\_translations if available, or falling back to the primary English fields. It should also decide whether to merge the base English text into the all\_translations object if an 'en' entry isn't explicitly in the translations table.
* **Data Freshness**: As a standard view, it always reflects the current state of the underlying tables. If it were materialized, a refresh strategy would be needed.

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

* The view's scalability is tied to the performance of the underlying tables and the translations table, particularly the efficiency of the translation aggregation subquery.
* Adding new translatable fields to accommodation\_types\_master would require updating this view definition to include them in the jsonb\_build\_object within all\_translations.

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

* 🔴 **Create View**: Execute the DDL to create public.v\_accommodation\_types\_localized.
* 🔴 **Verify Index on** translations: Ensure the recommended composite index on public.translations exists for optimal performance.
* 🟠 **API Layer Integration**: Plan how the API layer will consume this view, particularly how it will select the appropriate language from the all\_translations JSONB object or use the base fields.
* 🟢 **Testing**: Thoroughly test view performance, especially with a significant number of languages and translations.
* 🟢 **Documentation**: Document this view in the overall database schema and for API developers.