<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/34bd37922031b95a

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

4.32 Accommodation Meal Services Table (Version 1.4)

----------------------------------------------------

This document details the structure, purpose, and considerations for the `accommodation\_meal\_services` table, which links accommodations to the specific meal services they provide, including pricing and timing details. Version 1.4 adds a point to "Future Considerations" regarding parent table timestamp updates.

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

The `accommodation\_meal\_services` table is a junction (linking) table that connects accommodations to the specific meal services they provide (defined in `meal\_services\_master`). It allows each accommodation to list multiple meal services and, crucially, to provide context-specific details such as pricing, timing, and other notes for each service offered.

Key user-story touchpoints:

- Pilgrim (Anna): Seeing detailed meal information for an accommodation, including what meals are offered (e.g., "Breakfast," "Pilgrim Dinner"), their prices, and any specific notes like "Dinner must be booked by noon." (Story A3)

- Pilgrim (Anna): Filtering or making choices based on meal availability and cost.

- Accommodation Host (Marco): Specifying which meals their B&amp;B offers, setting prices for them, and adding important details about availability or what's included. (Story B1)

- System/UI: Dynamically constructing the list of offered meal services with their specific details for display on an accommodation's page.

### 2\. Schema (Markdown Table)

\*(No change to column structure from Version 1.3)\*

| column | data\_type | constraints | description |

| id | `uuid` | Primary Key (Default: `gen\_random\_uuid()`) | Unique identifier for this specific accommodation-meal service link. |

| accommodation\_waypoint\_id | `bigint` | Not Null, Foreign Key to `accommodations(waypoint\_id)` ON DELETE CASCADE, Part of UNIQUE constraint `uq\_accommodation\_meal\_service` | Links to the specific accommodation. |

| meal\_service\_id | `integer` | Not Null, Foreign Key to `meal\_services\_master(id)` ON DELETE RESTRICT, Part of UNIQUE constraint `uq\_accommodation\_meal\_service` | Links to the specific meal service from the master list. RESTRICT prevents deleting a master service if in use. |

| price\_amount | `numeric(10, 2)` | Nullable, CHECK (`price\_amount` IS NULL OR `price\_amount` >= 0) | Price for this meal service at this accommodation. Numeric for precision. |

| price\_currency\_code | `char(3)` | Not Null, Default 'EUR', CHECK (length(`price\_currency\_code`) = 3 AND `price\_currency\_code` = upper(`price\_currency\_code`)) | ISO 4217 currency code for `price\_amount` (e.g., "EUR"). V1 defaults to and requires EUR. |

| price\_is\_per\_person | `boolean` | Not Null, Default true | Indicates if `price\_amount` is per person (common for meals). Set to false if the price is, for example, for a family-style shared dish not priced individually. |

| availability\_and\_timing\_notes | `text` | Nullable | Details like specific times available, what's included (e.g., "Breakfast 07:00-09:30, continental"), booking requirements. Primary reference language (English) text. Translatable. |

| created\_at | `timestamp with time zone` | Not Null, Default `now()` | Timestamp of when this meal service was linked to the accommodation. |

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

| updated\_at | `timestamp with time zone` | Not Null, Default `now()` | Timestamp of when details like price or notes were last updated (auto-updated by trigger). |

| updated\_by\_profile\_id | `uuid` | Nullable, Foreign Key to `profiles(id)` ON DELETE SET NULL | Profile ID of the user who last updated details for this linked meal service. |

### 3\. PostgreSQL DDL

\*(DDL for table structure, comments, main trigger, and existing indexes remain the same as Version 1.3. Only the version in the table comment changes.)\*

SQL

```

-- Assumes public.accommodations, public.meal\_services\_master, public.profiles tables exist

-- Assumes public.set\_current\_timestamp\_updated\_at() function exists

-- Assumes public.cleanup\_related\_translations(TEXT, TEXT) function exists for the AFTER DELETE trigger

CREATE TABLE public.accommodation\_meal\_services (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

accommodation\_waypoint\_id BIGINT NOT NULL,

meal\_service\_id INTEGER NOT NULL,

price\_amount NUMERIC(10, 2) NULL CHECK (price\_amount IS NULL OR price\_amount >= 0),

price\_currency\_code CHAR(3) NOT NULL DEFAULT 'EUR' CHECK (length(price\_currency\_code) = 3 AND price\_currency\_code = upper(price\_currency\_code)),

price\_is\_per\_person BOOLEAN NOT NULL DEFAULT TRUE,

availability\_and\_timing\_notes TEXT 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 uq\_accommodation\_meal\_service UNIQUE (accommodation\_waypoint\_id, meal\_service\_id),

CONSTRAINT fk\_accommodation\_waypoint

FOREIGN KEY(accommodation\_waypoint\_id)

REFERENCES public.accommodations(waypoint\_id)

ON DELETE CASCADE,

CONSTRAINT fk\_meal\_service

FOREIGN KEY(meal\_service\_id)

REFERENCES public.meal\_services\_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.accommodation\_meal\_services IS 'Junction table linking accommodations to meal services they offer, with pricing and availability details. Version 1.4';

COMMENT ON COLUMN public.accommodation\_meal\_services.id IS 'Unique identifier for the accommodation-meal service link.';

COMMENT ON COLUMN public.accommodation\_meal\_services.accommodation\_waypoint\_id IS 'FK to accommodations(waypoint\_id). Part of a unique constraint uq\_accommodation\_meal\_service.';

COMMENT ON COLUMN public.accommodation\_meal\_services.meal\_service\_id IS 'FK to meal\_services\_master(id). Part of a unique constraint uq\_accommodation\_meal\_service.';

COMMENT ON COLUMN public.accommodation\_meal\_services.price\_amount IS 'Price for this meal service at this accommodation. Numeric(10,2).';

COMMENT ON COLUMN public.accommodation\_meal\_services.price\_currency\_code IS 'ISO 4217 currency code for price\_amount. V1 defaults to and requires "EUR".';

COMMENT ON COLUMN public.accommodation\_meal\_services.price\_is\_per\_person IS 'Indicates if price\_amount is per person (true) or for the service unit (false).';

COMMENT ON COLUMN public.accommodation\_meal\_services.availability\_and\_timing\_notes IS 'Details like times available, what''s included, booking requirements. Primary reference language (English) text. Translatable via ''translations'' table using this record''s `id` as row\_foreign\_key.';

COMMENT ON COLUMN public.accommodation\_meal\_services.created\_at IS 'Timestamp of when this meal service was linked to the accommodation.';

COMMENT ON COLUMN public.accommodation\_meal\_services.created\_by\_profile\_id IS 'Profile ID of the user who linked this meal service.';

COMMENT ON COLUMN public.accommodation\_meal\_services.updated\_at IS 'Timestamp of when details like price or notes were last updated.';

COMMENT ON COLUMN public.accommodation\_meal\_services.updated\_by\_profile\_id IS 'Profile ID of the user who last updated details for this link.';

-- Trigger for updated\_at

CREATE TRIGGER trigger\_accommodation\_meal\_services\_set\_updated\_at

BEFORE UPDATE ON public.accommodation\_meal\_services

FOR EACH ROW

WHEN (

OLD.price\_amount IS DISTINCT FROM NEW.price\_amount OR

OLD.price\_currency\_code IS DISTINCT FROM NEW.price\_currency\_code OR

OLD.price\_is\_per\_person IS DISTINCT FROM NEW.price\_is\_per\_person OR

OLD.availability\_and\_timing\_notes IS DISTINCT FROM NEW.availability\_and\_timing\_notes

)

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

COMMENT ON TRIGGER trigger\_accommodation\_meal\_services\_set\_updated\_at ON public.accommodation\_meal\_services IS 'Trigger to automatically update updated\_at timestamp when relevant details are modified.';

-- Trigger for orphan translation cleanup

CREATE OR REPLACE FUNCTION public.cleanup\_accommodation\_meal\_services\_translations()

RETURNS TRIGGER AS $$

BEGIN

IF TG\_OP = 'DELETE' THEN

DELETE FROM public.translations

WHERE table\_identifier = 'accommodation\_meal\_services'

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

END IF;

RETURN OLD;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER trigger\_cleanup\_accommodation\_meal\_services\_translations

AFTER DELETE ON public.accommodation\_meal\_services

FOR EACH ROW

EXECUTE FUNCTION public.cleanup\_accommodation\_meal\_services\_translations();

COMMENT ON TRIGGER trigger\_cleanup\_accommodation\_meal\_services\_translations ON public.accommodation\_meal\_services IS 'Cleans up orphaned translations from public.translations when an accommodation\_meal\_services record is deleted.';

-- Indexes

CREATE INDEX idx\_acc\_meal\_services\_meal\_service\_id ON public.accommodation\_meal\_services(meal\_service\_id);

COMMENT ON INDEX public.idx\_acc\_meal\_services\_meal\_service\_id IS 'Index to efficiently find all accommodations offering a specific meal service.';

CREATE INDEX IF NOT EXISTS idx\_ams\_created\_by\_profile\_id ON public.accommodation\_meal\_services(created\_by\_profile\_id) WHERE created\_by\_profile\_id IS NOT NULL;

CREATE INDEX IF NOT EXISTS idx\_ams\_updated\_by\_profile\_id ON public.accommodation\_meal\_services(updated\_by\_profile\_id) WHERE updated\_by\_profile\_id IS NOT NULL;

```

### 4\. JSON Schema Mirror

\*(No change from Version 1.3, as column structure is the same)\*

JSON

```

{

"title": "accommodation\_meal\_service\_link",

"description": "Links an accommodation to a specific meal service offered, with pricing and availability details. Version 1.4",

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid",

"description": "Unique identifier for this link. Primary Key.",

"readOnly": true

},

"accommodation\_waypoint\_id": {

"type": "integer",

"format": "int64",

"description": "FK to accommodations(waypoint\_id). Part of a unique constraint with meal\_service\_id."

},

"meal\_service\_id": {

"type": "integer",

"description": "FK to meal\_services\_master(id). Part of a unique constraint with accommodation\_waypoint\_id."

},

"price\_amount": {

"type": ["number", "null"],

"format": "float",

"description": "Price for this meal service at this accommodation.",

"minimum": 0

},

"price\_currency\_code": {

"type": "string",

"minLength": 3,

"maxLength": 3,

"pattern": "^[A-Z]{3}$",

"default": "EUR",

"description": "ISO 4217 currency code for price\_amount. V1 defaults to and requires EUR."

},

"price\_is\_per\_person": {

"type": "boolean",

"default": true,

"description": "Indicates if price\_amount is per person."

},

"availability\_and\_timing\_notes": {

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

"description": "Details like times available, what's included, booking requirements. Primary reference language (English) text. Translatable."

},

"created\_at": {

"type": "string",

"format": "date-time",

"readOnly": true

},

"created\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who linked this meal service."

},

"updated\_at": {

"type": "string",

"format": "date-time",

"readOnly": true

},

"updated\_by\_profile\_id": {

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

"format": "uuid",

"description": "Profile ID of the user who last updated details for this link."

}

},

"required": [

"accommodation\_waypoint\_id",

"meal\_service\_id",

"price\_currency\_code",

"price\_is\_per\_person"

],

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

"uniqueKeys": [

{"name": "uq\_accommodation\_meal\_service", "columns": ["accommodation\_waypoint\_id", "meal\_service\_id"]}

]

}

```

### 5\. Relationships & Integrity

\*(No change from Version 1.3)\*

- Primary Key: `id` (UUID, surrogate key).

- Unique Constraint: `uq\_accommodation\_meal\_service` on (`accommodation\_waypoint\_id`, `meal\_service\_id`).

- Foreign Keys: As defined in Version 1.3.

- Mermaid ER Snippet: (As provided in Version 1.3 documentation).

### 6\. Multilingual Strategy

\*(No change from Version 1.3)\*

- `availability\_and\_timing\_notes`: Translatable.

- Translation Management: Via `public.translations` table (using `id` as `row\_foreign\_key`) and orphan cleanup trigger.

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

\*(No change from Version 1.3)\*

- Key Fields for Workflow: `price\_amount`, `availability\_and\_timing\_notes` managed by hosts or admins/managers. Audit fields track links.

- RLS Policies: Public read for published accommodations; hosts manage their own.

### 8\. ENUM vs Lookup Discussion

\*(No change from Version 1.3)\*

- Not applicable directly. Links to `meal\_services\_master` (a lookup table).

### 9\. UI/UX Enablement

\*(No change from Version 1.3)\*

- Enables UI to list specific meal services, prices, and notes.

### 10\. Key Considerations & Definitions

\*(No change from Version 1.3)\*

- `ON DELETE RESTRICT` for `meal\_service\_id`: Important.

- Currency Handling: `price\_currency\_code` is `NOT NULL DEFAULT 'EUR'`.

- `price\_is\_per\_person` default: `true` is common for pilgrim meals.

### 11\. Scalability & Future-Proofing

\*(Section updated to include the new consideration)\*

- Standard Junction Table: A scalable pattern.

- Pricing Flexibility: Separate `price\_amount` and `price\_currency\_code` allow for future multi-currency support.

- Parent Table Timestamp Updates (V2+): Consider if adding or removing meal services (INSERT/DELETE operations on this table), or significant updates beyond those covered by the current conditional `updated\_at` trigger (e.g., if other fields were added that signify a major change), should also trigger an update to the `updated\_at` timestamp of the parent `public.accommodations` record. This could be relevant for data synchronization or cache invalidation strategies for the parent accommodation. Implementing this would require additional trigger logic or modifying the existing `updated\_at` trigger's `WHEN` condition.

### 12\. Next-Action Checklist

\*(No DDL changes required for this specific update, only documentation.)\*

- 🟢 Documentation: Ensure this "Future Consideration" regarding parent table timestamp updates is noted in this version of the specification document.

- 🟢 Verify Other Aspects: Confirm all other aspects from Version 1.3 (triggers, RLS policies, translation strategy using `id` as `row\_foreign\_key`) are correctly implemented and tested.