Database Schema Documentation: Core Entities & Data Flow

📋 Overview

This document covers the core entities in the meal delivery system: Orders, User Subscriptions, Meals, Items, and their relationships.

🏪 Core Entities

1. Orders (`orders`)

Purpose: Manages customer orders and delivery information

Structure

- **Primary Key**: `id` (UUID)
- **Key Fields**:
  - `user_id`, `subscription_id`, `order_number` (unique)
  - Financial: `subtotal`, `tax_amount`, `discount_amount`, `delivery_fee`, `total_amount`
  - Status: `status` (enum), `payment_status`, `payment_method`
  - Delivery: `delivery_address_id`, `scheduled_delivery_date`, `actual_delivery_date`

Constraints & Validation

- **Financial Integrity**: Multiple CHECK constraints ensuring non-negative amounts
- **Payment Status**: Limited to predefined states (`pending`, `paid`, `failed`, `refunded`, `partial_refund`)
- **Total Validation**: Ensures `total_amount = (subtotal + tax_amount + delivery_fee) - discount_amount`

Indexes

- **Performance**: `idx_orders_user_created`, `idx_orders_status_created`, `idx_orders_delivery_date`
- **Business Logic**: `idx_orders_payment_status` (filtered on paid status)
- **Unique**: `orders_order_number_key` for order tracking

Triggers & Data Flow

- **`trigger_set_order_number`**: Auto-generates sequential order numbers
- **`trigger_order_delivery`**: Updates subscription meal counts when order is delivered
- **`trigger_order_delivery_dates`**: Sets delivery dates based on subscription preferences
- **`trigger_update_timestamp_orders`**: Standard timestamp updates

---

2. User Subscriptions (`user_subscriptions`)

Purpose: Manages recurring meal subscription plans

Structure

- **Primary Key**: `id` (UUID)
- **Key Relationships**: `user_id`, `plan_id`, `delivery_address_id`, `payment_method_id`
- **Subscription Details**: `status`, `start_date`, `end_date`, `price_per_meal`, `total_meals`
- **Progress Tracking**: `consumed_meals`, `preferred_delivery_time`
- **Meal Planning**: `meals` (JSONB array of meal configurations)

Constraints

- **Date Validation**: `end_date > start_date`
- **Meal Limits**: `consumed_meals <= jsonb_array_length(meals)`
- **Financial**: Non-negative price validation

Indexes

- **User Management**: `idx_subscriptions_user_status` (user queries)
- **Active Subscriptions**: `idx_subscriptions_status_active` (filtered index)

Triggers & Business Logic

- **`trigger_validate_subscription`**: Validates subscription data before save
- **Integration with Orders**: Subscription consumption updated via order delivery trigger

---

3. Meals (`meals`)

Purpose: Central meal catalog with nutritional and dietary information

Structure

- **Primary Key**: `id` (serial integer)
- **Content**: Multi-language names/descriptions (`name`, `name_arabic`, etc.)
- **Nutritional**: `calories`, `protein_g`, `carbs_g`, `fat_g`, `fiber_g`, `sugar_g`
- **Dietary Flags**: `is_vegetarian`, `is_vegan`, `is_gluten_free`, `is_dairy_free`
- **Business Logic**: `rating`, `rating_count`, `discount_percentage`, `is_available`

Constraints

- **Nutritional Validation**: All nutritional values >= 0
- **Rating Bounds**: `rating` between 0-5, `spice_level` 0-5
- **Discount Validation**: `discount_percentage` 0-100

Indexes

- **Search & Filtering**:
  - `idx_meals_rating` (descending for popular meals)
  - `idx_meals_dietary` (dietary restriction filtering)
  - `idx_meals_available_featured` (active catalog)
  - `idx_meals_discount` (promotional queries)

Triggers

- `trigger_update_timestamp_meals`: Standard timestamp updates

---

4. Items (`items`)

Purpose: Individual food components/additives that compose meals

Structure

- **Primary Key**: `id` (serial integer)
- **Content**: Multi-language names/descriptions
- **Nutritional**: Similar to meals but simpler (`calories`, `protein_g`, `carbs_g`, `fat_g`)
- **Inventory**: `is_available`, `max_free_per_meal`, `is_additive`

Constraints

- **Nutritional Validation**: All values >= 0
- **Pricing**: `price >= 0`

Triggers

- `trigger_update_timestamp_items`: Standard timestamp updates

🔗 Junction Tables & Relationships

1. Order Composition

```
orders
├── order_meals (1:N)
│   └── order_items (1:N)
└── order_items (direct 1:N)
```

`order_meals`:

- Links orders to meal selections
- Stores meal-specific pricing and customization
- **Key**: `order_id`, `meal_id` with quantity and pricing

`order_items`:

- Individual items within orders (additives/sides)
- Can be linked to `order_meals` or directly to orders
- **Key**: `order_id`, `item_id` with quantity and unit pricing

2. Meal Composition

`meal_items`:

- Defines which items compose a meal using integer arrays

- **Index**: GIN index on `items` array for fast lookups
- **Trigger**: `trg_validate_meal_items_array` validates item existence

3. Subscription Planning

`plan_meals`:

- Links subscription plans to available meals
- **Features**: `is_substitutable` flag for meal flexibility

4. Allergy Management

`meal_allergies` & **`item_allergies`**:

- Many-to-many relationships for allergy tracking
- Cascade deletes for maintenance

📊 Data Flow & Business Logic

Order → Subscription Integration

1. **Order Creation**: `trigger_set_order_number` generates unique order numbers
2. **Delivery Processing**: `trigger_order_delivery` updates subscription meal counts
3. **Subscription Sync**: Increments `consumed_meals` and cycles `next_delivery_meal`

Rating System

1. **Review Creation**: `meal_reviews` triggers update meal ratings
2. **Real-time Aggregation**: `trigger_update_meal_rating_insert/update` recalculates averages
3. **Published Filter**: Only published reviews affect public ratings

Meal Validation Pipeline

1. **Array Validation**: `validate_meal_items_array()` ensures meal items exist
2. **Nutritional Integrity**: CHECK constraints maintain data quality
3. **Availability Filtering**: Indexed queries for active meals

---

🎯 Key Design Patterns

1. Multi-language Support

- Consistent `_arabic` suffix for localized content
- Language preferences in user profiles

2. Temporal Tracking

- Consistent `created_at`, `updated_at` across all tables
- Automated via `update_timestamp()` function

3. Financial Integrity

- Multiple CHECK constraints for amount validation
- Calculated totals with validation

4. Subscription Mechanics

- JSONB `meals` array for flexible meal planning
- Consumption tracking with bounds checking
- Delivery scheduling based on preferences

5. Allergy & Dietary Management

- Comprehensive many-to-many relationships
- User-level overrides for allergy severity
- Cascade deletion for maintainability

This schema supports a sophisticated meal delivery system with subscription management, detailed nutritional tracking, multi-language support, and complex order composition workflows.