

FreshCart Database

This document provides a comprehensive overview of all database entities in the FreshCart system, including their attributes and relationships.

Table of Contents

- 1. [User Entity](#)
- 2. [Profile Entity](#)
- 3. [Address Entity](#)
- 4. [Store Entity](#)
- 5. [Category Entity](#)
- 6. [Product Entity](#)
- 7. [Inventory Entity](#)
- 8. [Order Entity](#)
- 9. [OrderItem Entity](#)
- 10. [Payment Entity](#)
- 11. [Entity Relationships Diagram](#)

User Entity

Table: `user`

Attributes

Field	Type	Constraints	Description
<code>user_id</code>	<code>number</code>	Primary Key, Auto-increment	Unique identifier for the user
<code>profile_id</code>	<code>number</code>	Unique, Not Null	Foreign key to Profile entity
<code>email</code>	<code>string(255)</code>	Unique, Not Null	User's email address
<code>password</code>	<code>string(255)</code>	Not Null	Hashed password
<code>role</code>	<code>Role</code> (enum)	Not Null	User role: Admin, Customer, Store, Driver
<code>hashedRefreshToken</code>	<code>string(255)</code>	Nullable	JWT refresh token (hashed)
<code>createdAt</code>	<code>timestamp</code>	Default: CURRENT_TIMESTAMP	Account creation date
<code>updatedAt</code>	<code>timestamp</code>	Auto-update	Last modification date

Relationships

- **One-to-One:** Profile (`user.profile_id` → `profile.profile_id`)

- **One-to-Many:** Orders (user.user_id ← order.user_id)
- **One-to-Many:** Stores (user.user_id ← store.owner_id)

Enums

```
enum Role {
  Admin = 'Admin',
  Customer = 'Customer',
  Store = 'Store',
  Driver = 'Driver'
}
```

Profile Entity

Table: profile

Attributes

Field	Type	Constraints	Description
profile_id	number	Primary Key, Auto-increment	Unique identifier for the profile
first_name	string(255)	Not Null	User's first name
last_name	string(255)	Not Null	User's last name
phone_number	string(20)	Nullable	User's phone number
town	string(255)	Nullable	User's town
county	string(255)	Nullable	User's county
country	string(255)	Nullable	User's country
createdAt	timestamp	Default: CURRENT_TIMESTAMP	Profile creation date
updatedAt	timestamp	Auto-update	Last modification date

Relationships

- **One-to-One:** User (profile.profile_id ← user.profile_id)
- **One-to-Many:** Addresses (profile.profile_id ← address.profile_id)

Address Entity

Table: address

Attributes

Field	Type	Constraints	Description
-------	------	-------------	-------------

Field	Type	Constraints	Description
address_id	number	Primary Key, Auto-increment	Unique identifier for the address
profile_id	number	Foreign Key, Not Null	Reference to Profile entity
street	string(100)	Nullable	Street address
town	string(100)	Not Null	Town name
county	string(100)	Not Null	County name
postal_code	string(20)	Not Null	Postal/ZIP code
country	string(100)	Default: 'Kenya'	Country name
type	string (enum)	Default: 'home'	Address type: home, work, other
isDefault	boolean	Default: false	Whether this is the default address
created_at	timestamp	Default: CURRENT_TIMESTAMP	Creation date
updated_at	timestamp	Auto-update	Last modification date

Relationships

- **Many-to-One:** Profile (address.profile_id → profile.profile_id)

Store Entity

Table: store

Attributes

Field	Type	Constraints	Description
store_id	number	Primary Key, Auto-increment	Unique identifier for the store
owner_id	number	Unique, Foreign Key	Reference to User entity (store owner)
name	string(255)	Not Null	Store name
description	string(255)	Not Null	Store description
city	string(100)	Nullable	Store city
town	string(100)	Nullable	Store town
country	string(100)	Default: 'Kenya'	Store country
contact_info	string(255)	Not Null	Store contact information
image_url	string(255)	Nullable	Store image URL
rating	decimal(3,2)	Default: 0	Store rating (0.00-5.00)

Field	Type	Constraints	Description
total_reviews	number	Default: 0	Total number of reviews
store_code	string(50)	Unique	Unique store code
delivery_fee	number	Default: 0	Store delivery fee
created_at	timestamp	Default: CURRENT_TIMESTAMP	Store creation date
updated_at	timestamp	Auto-update	Last modification date

Relationships

- **Many-to-One:** User (store.owner_id → user.user_id)
- **One-to-Many:** Orders (store.store_id ← order.store_id)
- **One-to-Many:** Inventories (store.store_id ← inventory.store_id)

Category Entity

Table: category

Attributes

Field	Type	Constraints	Description
category_id	number	Primary Key, Auto-increment	Unique identifier for the category
name	string(255)	Not Null	Category name
description	string(255)	Not Null	Category description
image_url	string(500)	Nullable	Category image URL
created_at	timestamp	Default: CURRENT_TIMESTAMP	Category creation date

Relationships

- **One-to-Many:** Products (category.category_id ← product.category_id)

Product Entity

Table: product

Attributes

Field	Type	Constraints	Description
product_id	number	Primary Key, Auto-increment	Unique identifier for the product

Field	Type	Constraints	Description
category_id	number	Foreign Key, Not Null	Reference to Category entity
name	string(255)	Not Null	Product name
description	string(255)	Not Null	Product description
price	decimal(10,2)	Not Null	Product price
stock_quantity	number	Not Null	Available stock quantity
image_url	string(255)	Nullable	Product image URL
weight	decimal(8,3)	Nullable	Product weight in kg
unit	string(50)	Nullable	Unit of measurement (kg, pieces, liters)
rating	decimal(3,2)	Default: 0	Product rating (0.00-5.00)
review_count	number	Default: 0	Number of reviews
discount	number	Default: 0	Discount percentage
expiry_date	date	Nullable	Product expiry date
created_at	timestamp	Default: CURRENT_TIMESTAMP	Product creation date
updatedAt	timestamp	Auto-update	Last modification date

Relationships

- **Many-to-One:** Category (product.category_id → category.category_id)
- **Many-to-Many:** Orders (product.product_id ↔ order.order_id via order_items)
- **Many-to-Many:** Inventory (product.product_id ↔ inventory.inventory_id via inventory_products)

Inventory Entity

Table: inventory

Attributes

Field	Type	Constraints	Description
inventory_id	number	Primary Key, Auto-increment	Unique identifier for inventory record
store_id	number	Foreign Key, Not Null	Reference to Store entity
stock_qty	number	Default: 0	Current stock quantity
quantity_reserved	number	Default: 0	Reserved quantity for pending orders

Field	Type	Constraints	Description
reorder_level	number	Default: 5	Minimum stock level before reorder alert
max_stock_level	number	Default: 100	Maximum stock capacity
cost_price	decimal(10,2)	Nullable	Cost price of the inventory item
last_restocked	date	Nullable	Date of last restock
last_action	InventoryAction (enum)	Default: RESTOCK	Last inventory action performed
created_at	timestamp	Default: CURRENT_TIMESTAMP	Inventory record creation date

Computed Properties

- total_quantity: stock_qty + quantity_reserved
- is_low_stock: stock_qty <= reorder_level
- is_out_of_stock: stock_qty === 0

Relationships

- Many-to-One:** Store (inventory.store_id → store.store_id)
- Many-to-Many:** Products (inventory.inventory_id ↔ product.product_id via inventory_products)

Enums

```
enum InventoryAction {
  RESTOCK = 'restock',
  SALE = 'sale',
  ADJUSTMENT = 'adjustment',
  EXPIRED = 'expired',
  DAMAGED = 'damaged'
}
```

Order Entity

Table: order

Attributes

Field	Type	Constraints	Description
order_id	number	Primary Key, Auto-increment	Unique identifier for the order

Field	Type	Constraints	Description
order_number	string(20)	Unique	Human-readable order number
user_id	number	Foreign Key, Not Null	Reference to User entity (customer)
store_id	number	Foreign Key, Not Null	Reference to Store entity
delivery_fee	decimal(10,2)	Default: 0	Delivery fee amount
discount_amount	decimal(10,2)	Default: 0	Discount applied to the order
status	OrderStatus (enum)	Default: PENDING	Current order status
delivery_method	DeliveryMethod (enum)	Default: STANDARD_DELIVERY	Delivery method
total_amount	decimal(10,2)	Not Null	Total order amount
estimated_delivery_time	timestamp	Nullable	Estimated delivery time
created_at	timestamp	Default: CURRENT_TIMESTAMP	Order creation date
delivery_address	string(255)	Not Null	Delivery address
tax_amount	number	Nullable	Tax amount
driver_id	number	Nullable	Reference to User entity (driver)
confirmed_at	timestamp	Nullable	Order confirmation timestamp
prepared_at	timestamp	Nullable	Order preparation completion timestamp
picked_up_at	timestamp	Nullable	Order pickup timestamp
delivered_at	timestamp	Nullable	Order delivery timestamp
cancelled_at	timestamp	Nullable	Order cancellation timestamp
cancellation_reason	text	Nullable	Reason for order cancellation
review	text	Nullable	Customer review of the order
rating	decimal(3,2)	Nullable	Customer rating (0.00-5.00)

Relationships

- **Many-to-One:** User/Customer (order.user_id → user.user_id)
- **Many-to-One:** Store (order.store_id → store.store_id)
- **Many-to-One:** User/Driver (order.driver_id → user.user_id)
- **Many-to-Many:** Products (order.order_id ↔ product.product_id via order_items)
- **One-to-Many:** Payments (order.order_id ← payment.order_id)
- **One-to-Many:** OrderItems (order.order_id ← order_item.order_id)

Enums

```
enum OrderStatus {
    PENDING = 'pending',
    CONFIRMED = 'confirmed',
    PREPARING = 'preparing',
    READY_FOR_PICKUP = 'ready_for_pickup',
    IN_TRANSIT = 'in_transit',
    DELIVERED = 'delivered',
    CANCELLED = 'cancelled',
    REFUNDED = 'refunded'
}

enum DeliveryMethod {
    PICKUP = 'pickup',
    STANDARD_DELIVERY = 'standard_delivery',
    EXPRESS_DELIVERY = 'express_delivery'
}
```

OrderItem Entity

Table: `order_item` (Join table for Order-Product relationship)

Attributes

Field	Type	Constraints	Description
<code>order_item_id</code>	<code>number</code>	Primary Key, Auto-increment	Unique identifier for the order item
<code>order_id</code>	<code>number</code>	Foreign Key, Not Null	Reference to Order entity
<code>product_id</code>	<code>number</code>	Foreign Key, Not Null	Reference to Product entity
<code>quantity</code>	<code>number</code>	Not Null	Quantity of the product ordered
<code>unit_price</code>	<code>decimal(10,2)</code>	Not Null	Unit price at time of order
<code>total_price</code>	<code>decimal(10,2)</code>	Not Null	Total price for this line item

Relationships

- **Many-to-One:** Order (order_item.order_id → order.order_id)
- **Many-to-One:** Product (order_item.product_id → product.product_id)

Payment Entity

Table: payment

Attributes

Field	Type	Constraints	Description
payment_id	number	Primary Key, Auto-increment	Unique identifier for the payment
order_id	number	Foreign Key, Not Null	Reference to Order entity
email	string(255)	Not Null	Customer email for payment
authorization_url	string(255)	Nullable	Payment gateway authorization URL
user_id	number	Foreign Key, Not Null	Reference to User entity
amount	decimal(10,2)	Not Null	Payment amount
currency	string(3)	Default: 'KES'	Payment currency
payment_method	PaymentMethod (enum)	Not Null	Payment method used
gateway	PaymentGateway (enum)	Not Null	Payment gateway used
payment_reference	string(255)	Unique	Unique payment reference
status	PaymentStatus (enum)	Default: PENDING	Payment status
transaction_id	string(255)	Nullable	Gateway transaction ID
gateway_reference	string(255)	Nullable	Gateway reference number
gateway_response	json	Nullable	Gateway response data
metadata	json	Nullable	Additional payment metadata
failure_reason	text	Nullable	Reason for payment failure
refunded_amount	decimal(10,2)	Default: 0	Amount refunded
processed_at	timestamp	Nullable	Payment processing timestamp
failed_at	timestamp	Nullable	Payment failure timestamp
updated_at	timestamp	Auto-update	Last modification date

Relationships

- **Many-to-One:** Order (payment.order_id → order.order_id)
- **Many-to-One:** User (payment.user_id → user.user_id)

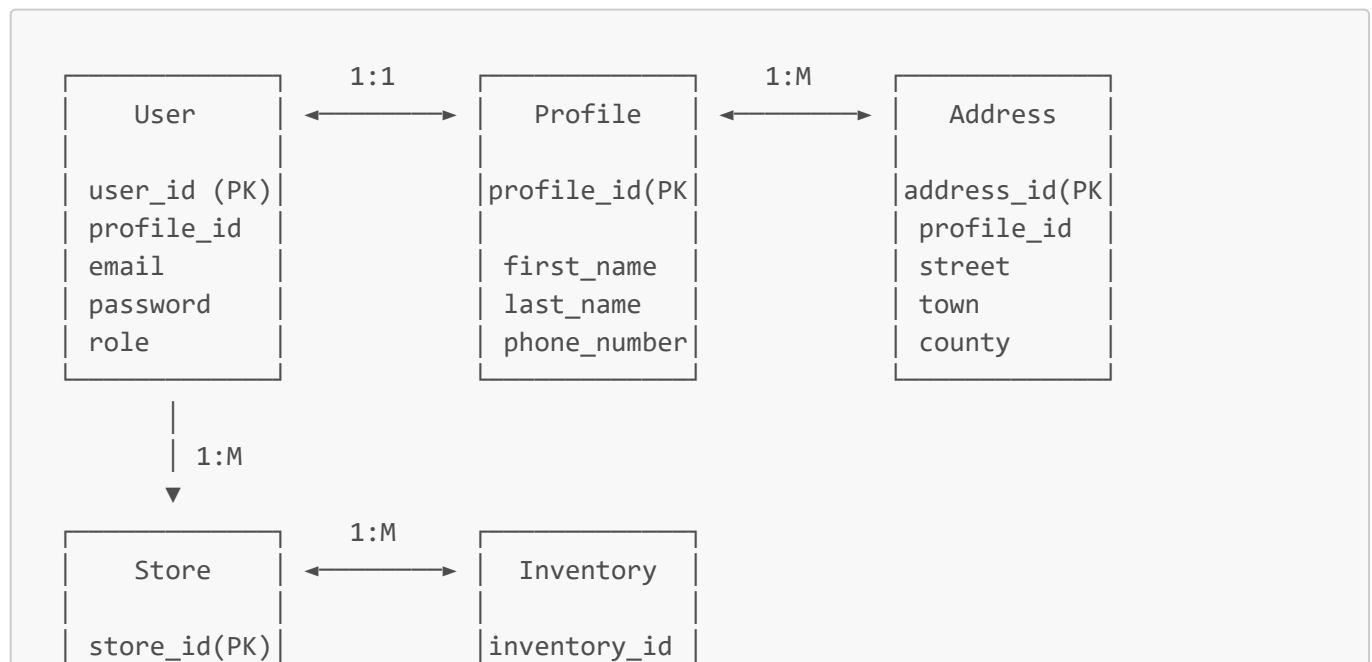
Enums

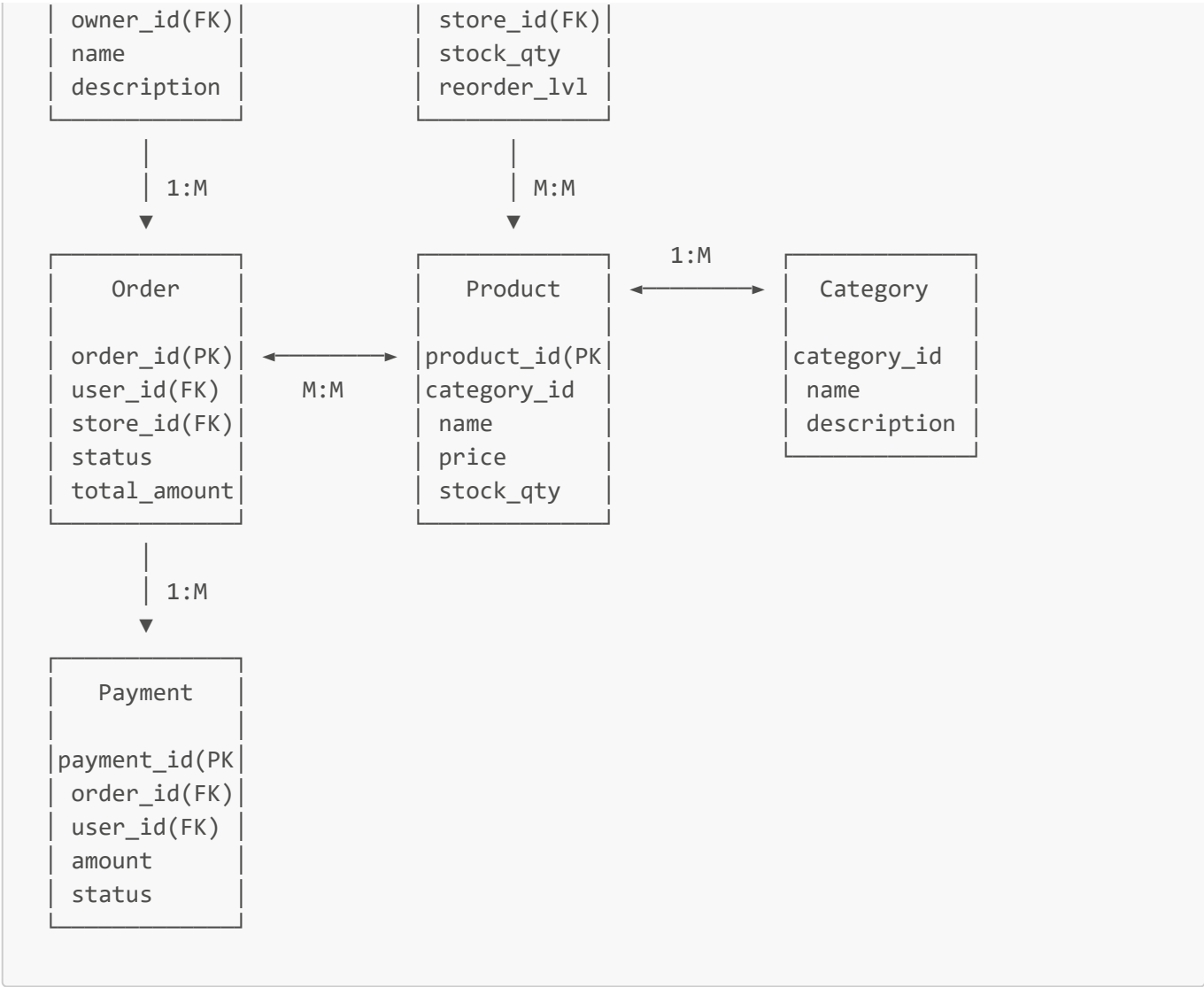
```
enum PaymentStatus {
    PENDING = 'pending',
    PROCESSING = 'processing',
    COMPLETED = 'completed',
    FAILED = 'failed',
    CANCELLED = 'cancelled',
    REFUNDED = 'refunded',
    PARTIALLY_REFUNDED = 'partially_refunded'
}

enum PaymentMethod {
    CARD = 'card',
    MOBILE_MONEY = 'mobile_money',
    CASH = 'cash',
    BANK_TRANSFER = 'bank_transfer'
}

enum PaymentGateway {
    STRIPE = 'stripe',
    PAYSTACK = 'paystack',
    MPESA = 'mpesa',
    PAYPAL = 'paypal',
    FLUTTERWAVE = 'flutterwave',
    CASH = 'cash'
}
```

Entity Relationships Diagram





Database Configuration

The application uses PostgreSQL with TypeORM for database management. Key configuration details:

- **Database Provider:** PostgreSQL (Neon Database)
- **ORM:** TypeORM
- **Synchronization:** Enabled in development
- **Migration Support:** Available
- **SSL:** Enabled with `rejectUnauthorized: false`

Module Dependencies

Each entity is organized into its own module with the following structure:

- **Entity Definition:** TypeORM entity with decorators
- **Service:** Business logic and data operations
- **Controller:** HTTP endpoint handlers
- **Module:** NestJS module configuration
- **DTOs:** Data Transfer Objects for validation

Notes

1. **Foreign Key Relationships:** All foreign key relationships are properly configured with TypeORM decorators.
2. **Cascade Operations:** Critical relationships include cascade delete to maintain data integrity.
3. **Computed Properties:** Some entities include computed properties for business logic (e.g., inventory stock calculations).
4. **Enums:** Consistent use of TypeScript enums for status fields and categorization.
5. **Timestamps:** All entities include creation and update timestamps for audit trails.
6. **Indexes:** Unique constraints and indexes are applied where necessary for performance and data integrity.

This documentation serves as a reference for developers working with the FreshCart backend database schema.