API Endpoints Description

🔄 Inventory Endpoints (/inventory)

- **GET /inventory/**
  Returns a list of all inventory items.
  Response: List[InventoryOut]
- **GET /inventory/low**
  Returns inventory items that are low in stock (e.g., below a predefined threshold).
  Response: List[InventoryOut]
- **POST /inventory/{product_id}**
  Updates the inventory for a specific product.
  Body: InventoryUpdate
  Response: InventoryOut
  Errors: 404 Not Found if inventory doesn't exist.
- **GET /inventory/history**
  Fetches the full history of inventory updates.
  Response: List[InventoryHistoryOut]
- **GET /inventory/{product_id}**
  Fetches inventory details for a specific product by product_id.
  Response: InventoryOut
  Errors: 404 Not Found if inventory doesn't exist.

🗂️ Category Endpoints (/categories)

- **POST /categories/**
  Creates a new category.
  Body: CategoryCreate
  Response: CategoryOut
- **GET /categories/**
  Lists all available categories.
  Response: List[CategoryOut]

- **GET /categories/{category_id}**
  Retrieves a category by ID.
  Response: CategoryOut
  Errors: 404 Not Found if category doesn't exist.

- **PUT /categories/{category_id}**
  Updates a category by ID.
  Body: CategoryUpdate
  Response: CategoryOut

💸 Sales Endpoints (/sales)

- **POST /sales/**
  Records a new sale.
  Body: SaleCreate
  Response: SaleOut

- **GET /sales/**
  Fetches all sales within a date range. Optional filters: product_id, category_id.
  Query Params:
  - start_date (required) — format: YYYY-MM-DD
  - end_date (required) — format: YYYY-MM-DD
  - product_id (optional)
  - category_id (optional)
    Response: List[SaleOut]

📊 Revenue Reporting

All revenue endpoints return grouped revenue reports for the specified date range.

- **GET /sales/revenue/weekly**
  Weekly revenue aggregation.

- **GET /sales/revenue/monthly**
  Monthly revenue aggregation.

- **GET /sales/revenue/annual**
  Annual revenue aggregation.

- **GET /sales/revenue/custom-range**
  Custom time range revenue summary (used when other groupings are not sufficient).

All revenue endpoints require:

- start_date (required) — format: YYYY-MM-DD
- end_date (required) — format: YYYY-MM-DD

Database Schema Documentation

1. product

Stores the basic information about products available in the system.

Column	Type	Description
id	INTEGER	Primary key, unique product identifier.
name	TEXT	Name of the product.
description	TEXT	Optional description of the product.
price	REAL	Unit price of the product.
category_id	INTEGER	Foreign key referencing category(id).
created_at	DATETIME	Timestamp when the product was created.
updated_at	DATETIME	Last updated timestamp.

Relationships:

- Many products belong to one category (category_id foreign key).

---

2. category

Represents a category/grouping for organizing products.

Column	Type	Description
id	INTEGER	Primary key. Unique category ID.

Column	Type	Description
name	TEXT	Name of the category.
description	TEXT	Optional description.

Relationships:

- One category can have many products.

---

3. sales

Captures every sale transaction made in the system.

Column	Type	Description
sale_id	INTEGER	Primary key. Unique sale record.
sale_date	DATETIME	Timestamp when the sale occurred.
product_id	INTEGER	Foreign key referencing product(id).
quantity	INTEGER	Quantity of the product sold.
amount_processed	REAL	Total amount processed for the sale.
revenue	REAL	Profit or income generated from the sale.
category_id	INTEGER	Foreign key referencing category(id). (denormalized for performance/reporting)
last_updated_at	DATETIME	Last time this record was updated. (optional)

Relationships:

- Each sale is tied to a product and indirectly to a category.

---

4. inventory

Tracks current inventory levels for each product.

Column	Type	Description
product_id	INTEGER	Foreign key referencing product(id), also acts as primary key.
quantity	INTEGER	Current stock quantity for the product.
last_updated	DATETIME	Last time inventory was updated.

Relationships:

- One-to-one with product.

---

5. inventory_history

Maintains a historical log of inventory changes (useful for audit trail).

Column	Type	Description
id	INTEGER	Primary key.
product_id	INTEGER	Foreign key referencing product(id).
change_quantity	INTEGER	Number of items added or removed.
timestamp	DATETIME	Time of inventory change.
operation	TEXT	Description of the operation (e.g., "sale", "restock").

Relationships:

- Many history entries per product.

Entity Relationships Summary:

category (1) ---------- (∞) product

product (1) ---------- (1) inventory

product (1) ---------- (∞) sales

product (1) ---------- (∞) inventory_history