E-commerce Admin Database Schema Documentation

Schema Overview

This schema models the backend of an e-commerce system's admin dashboard, focusing on product management, inventory tracking, and sales reporting.

Tables and Descriptions

1. categories

- **Purpose**: Stores product categories for classification (e.g., Electronics, Books).
- **Fields**:
  - id (INT, PK): Unique identifier.
  - name (VARCHAR): Category name.
  - description (TEXT): Details about the category.
- **Relationships**:
  - One-to-many with products: A category can have many products.

2. products

- **Purpose**: Contains detailed information about items for sale.
- **Fields**:
  - id (INT, PK): Unique product ID.
  - name (VARCHAR): Product name.
  - description (TEXT): Description of the product.
  - category_id (INT, FK → categories.id): Links product to its category.
  - price (DECIMAL): Unit price.
- **Relationships**:
  - Belongs to categories.
  - One-to-one with inventory (each product has one inventory record).
  - One-to-many with inventory_logs, sale_items.

3. inventory

- **Purpose**: Tracks current stock levels of each product.
- **Fields**:
  - product_id (INT, PK, FK → products.id): Same as product ID.
  - quantity (INT): Quantity in stock.
  - updated_at (DATETIME): Timestamp of the last update.
- **Relationships**:
  - One-to-one with products.

4. inventory_logs

- **Purpose**: Logs all inventory changes (e.g., restocks, sales).
- **Fields**:
  - id (INT, PK): Unique log entry.
  - product_id (INT, FK → products.id): Product affected.
  - quantity_change (INT): Quantity added or removed.
  - change_type (ENUM: 'INCREASE', 'DECREASE'): Nature of change.
  - created_at (DATETIME): When the change happened.
- **Relationships**:
  - Many-to-one with products.

5. sales

- **Purpose**: Records each completed sale transaction.
- **Fields**:
  - id (INT, PK): Unique sale ID.
  - sale_date (DATETIME): Date and time of the sale.
  - total_amount (DECIMAL): Total value of the sale.
- **Relationships**:
  - One-to-many with sale_items: A sale can contain multiple items.

6. sale_items

- **Purpose**: Itemized breakdown of what was sold in each sale.
- **Fields**:
  - id (INT, PK): Unique item line ID.
  - sale_id (INT, FK → sales.id): Associated sale.
  - product_id (INT, FK → products.id): Product sold.
  - quantity (INT): Quantity sold.
  - price_per_unit (DECIMAL): Price at the time of sale.
- **Relationships**:
  - Many-to-one with both sales and products.

Entity Relationships Summary

categories 1 ------< products products 1 ------1 inventory products 1 ------< inventory_logs products 1 ------< sale_items sales 1 ------< sale_items

---

Design Benefits

- **Modularity**: Each table has a single responsibility.
- **Normalization**: Avoids redundant data.
- **Auditability**: inventory_logs provide full traceability of stock changes.
- **Scalability**: The schema supports adding new product types, categories, and transaction data with minimal changes.