Inventory Management System

---

1. Entity Design

Here are the updated entities and their relationships:

- **User** : Manages login and role-based access control (Admin, Sales, Warehouse, Supplier).
- **Product** : Represents items in the inventory, including product details like price, quantity, SKU, etc.
- **Supplier** : Manages the details of suppliers for replenishing stock.
- **Category** : Categorizes products for better organization (e.g., Electronics, Furniture).
- **Order** : Represents customer orders, tracking which products were ordered and their status.
- **Transaction** : Represents inventory movements such as stock purchases, sales, returns, or adjustments.
- **Warehouse** : Represents different warehouse locations where products are stored.
- **Inventory** : Tracks the stock of products within specific warehouses.
- **Shipping** : Tracks shipment details for outgoing orders.
- **Return** : Handles product returns and updates inventory accordingly.
- **Pricing** : Represents pricing adjustments, discounts, and promotional pricing.

---

2. Detailed Table Design

User Table (`users`)

Stores information about users with role-based access control.

Column Name	Data Type	Description
`user_id`	INT (PK)	Unique user ID
`first_name`	VARCHAR(255)	User's first name
`last_name`	VARCHAR(255)	User's last name
`email`	VARCHAR(255)	User's email address
`password`	VARCHAR(255)	Encrypted password
`role`	ENUM('Admin', 'Manager', 'Warehouse', 'Sales', 'Supplier')	User's role
`created_at`	TIMESTAMP	Account creation timestamp
`updated_at`	TIMESTAMP	Last updated timestamp

Product Table (`products`)

Stores product details, including stock information.

Column Name	Data Type	Description
`product_id`	INT (PK)	Unique product ID
`name`	VARCHAR(255)	Product name
`description`	TEXT	Product description
`sku`	VARCHAR(100)	Stock Keeping Unit (SKU)
`price`	DECIMAL(10,2)	Product price
`category_id`	INT (FK)	Reference to product category
`supplier_id`	INT (FK)	Reference to product supplier
`created_at`	TIMESTAMP	Product creation timestamp
`updated_at`	TIMESTAMP	Last updated timestamp

Supplier Table (`suppliers`)

Stores supplier information.

Column Name	Data Type	Description
`supplier_id`	INT (PK)	Unique supplier ID
`name`	VARCHAR(255)	Supplier name
`contact_info`	TEXT	Supplier contact information
`created_at`	TIMESTAMP	Supplier creation timestamp

Category Table (`categories`)

Stores product categories (e.g., electronics, furniture, etc.).

Column Name	Data Type	Description
`category_id`	INT (PK)	Unique category ID
`name`	VARCHAR(255)	Category name
`description`	TEXT	Category description
`created_at`	TIMESTAMP	Category creation timestamp

Order Table (`orders`)

Stores customer order details.

Column Name	Data Type	Description

Column Name	Data Type	Description
`order_id`	INT (PK)	Unique order ID
`user_id`	INT (FK)	Reference to the user (customer)
`total_price`	DECIMAL(10,2)	Total price of the order
`status`	ENUM('Pending', 'Completed', 'Shipped', 'Returned')	Order status
`created_at`	TIMESTAMP	Order creation timestamp
`updated_at`	TIMESTAMP	Last updated timestamp

Transaction Table (`transactions`)

Tracks inventory movements such as sales, purchases, returns, and adjustments.

Column Name	Data Type	Description
`transaction_id`	INT (PK)	Unique transaction ID
`product_id`	INT (FK)	Reference to the product
`quantity`	INT	Quantity of items moved
`transaction_type`	ENUM('Sale', 'Purchase', 'Return', 'Adjustment')	Type of transaction
`created_at`	TIMESTAMP	Transaction timestamp

Warehouse Table (`warehouses`)

Stores information about different warehouse locations.

Column Name	Data Type	Description
`warehouse_id`	INT (PK)	Unique warehouse ID
`name`	VARCHAR(255)	Warehouse name
`location`	VARCHAR(255)	Warehouse location
`created_at`	TIMESTAMP	Warehouse creation timestamp

Inventory Table (`inventory`)

Stores stock levels of products in different warehouses.

Column Name	Data Type	Description
`inventory_id`	INT (PK)	Unique inventory record ID
`product_id`	INT (FK)	Reference to the product

Column Name	Data Type	Description
warehouse_id	INT (FK)	Reference to the warehouse
stock_qty	INT	Quantity of the product in stock
created_at	TIMESTAMP	Inventory record creation timestamp

Shipping Table (shipping)

Tracks shipment details for customer orders.

Column Name	Data Type	Description
shipping_id	INT (PK)	Unique shipping ID
order_id	INT (FK)	Reference to the order being shipped
tracking_number	VARCHAR(255)	Shipment tracking number
status	ENUM('Pending', 'Shipped', 'Delivered', 'Returned')	Shipping status
shipped_at	TIMESTAMP	Shipment date
created_at	TIMESTAMP	Shipping record creation timestamp

Return Table (returns)

Tracks product returns and updates stock accordingly.

Column Name	Data Type	Description
return_id	INT (PK)	Unique return ID
order_id	INT (FK)	Reference to the order being returned
product_id	INT (FK)	Reference to the returned product
quantity	INT	Quantity of the returned product
return_reason	TEXT	Reason for the return
created_at	TIMESTAMP	Return record creation timestamp

Pricing Table (pricing)

Stores product pricing adjustments (e.g., discounts, promotions).

Column Name	Data Type	Description

Column Name	Data Type	Description
pricing_id	INT (PK)	Unique pricing adjustment ID
product_id	INT (FK)	Reference to the product
discount	DECIMAL(10,2)	Discount amount
promotion	TEXT	Promotional offer
start_date	TIMESTAMP	Start date of the pricing adjustment
end_date	TIMESTAMP	End date of the pricing adjustment

---

3. Relationships Between Entities

- **One-to-Many** :
- A **user** can place multiple **orders** .
- A **category** can have multiple **products** .
- A **supplier** can supply multiple **products** .
- A **warehouse** can store multiple **products** (through the **inventory** table).
- **Many-to-Many** :
- A **product** can appear in many **orders** , and an **order** can have many **products** (this is handled by the **order_items** table, which is not explicitly shown here but would be created as an intermediate table).
- **One-to-One** :
- An **order** has one **shipping** record.
- A **return** can only relate to a single **order** and **product** .

---

4. Conclusion

This extended **Inventory Management System** (IMS) design includes all the necessary entities and relationships to handle a full range of

operations, from product management, inventory tracking, order handling, to shipping and returns. By breaking down the operations into modular entities, this design ensures scalability, flexibility, and a structured approach to managing inventory.