

P2 DATABASE DESIGN, INITIAL ERD

Group 4

Stock Market Management

Geeta Tirumalasetty

Suyeesha

Neil

Amreesh

Product/Item

Attributes:

ProductID (Primary Key)

Name

Description

Price

QuantityOnHand

ReorderPoint

SupplierID (Foreign Key)

CategoryID (Foreign Key)

Relationships:

Belongs to one Supplier

Belongs to one Category

Has many Sales Orders

Has many Purchase Orders

Located in many Inventory Locations

Inventory Location

Attributes:

LocationID (Primary Key)

Name

Description

Relationships:

Contains many Products

Supplier/Vendor

Attributes:

SupplierID (Primary Key)

Name

ContactInfo

Relationships:

Supplies many Products

Purchase Order (PO)

Attributes:

OrderID (Primary Key)

OrderDate

ExpectedDeliveryDate

TotalAmount

SupplierID (Foreign Key)

Relationships:

Places many Purchase Order Items

Sales Order

Attributes:

OrderID (Primary Key)

OrderDate

CustomerID (Foreign Key)

TotalAmount

Relationships:

Contains many Sales Order Items

Warehouse/Storage Space

Attributes:

WarehouseID (Primary Key)

Name

Address

Relationships:

Contains many Inventory Locations

Stock Keeping Unit (SKU)

Attributes:

SKU (Primary Key)

Barcode

ProductID (Foreign Key)

LocationID (Foreign Key)

Relationships:

Belongs to one Product

Belongs to one Inventory Location

Inventory Transaction

Attributes:

TransactionID (Primary Key)

TransactionDate

Type (e.g., Sale, Purchase, Adjustment)

Quantity

ProductID (Foreign Key)

LocationID (Foreign Key)

Relationships:

Involves one Product

Occurs at one Inventory Location

Product Category/Classification

Attributes:

CategoryID (Primary Key)

Name

Description

Relationships:

Contains many Products

Supplier Lead Time

Attributes:

SupplierID (Primary Key, Foreign Key)

LeadTime

INITIAL ERD

Creating an ERD (Entity-Relationship Diagram) for an Inventory Management system involves identifying the core entities that will form the basis of the database. The ERD is crucial for visualizing the structure of the database and understanding how various entities are interrelated. For an Inventory Management system, addressing business problems such as stock management, order processing, supplier management, and customer satisfaction is critical. Here's how you can approach this task:

Business Problems Being Addressed

The database aims to solve several business problems inherent in inventory management, including:

1. ***Stock Optimization***: Ensuring that stock levels are maintained at an optimal level to meet demand without overstocking.
2. ***Order Processing Efficiency***: Streamlining the process from order placement to fulfillment, reducing processing times and errors.
3. ***Supplier Management***: Managing relationships with suppliers to ensure timely restocking of inventory at competitive prices.
4. ***Customer Satisfaction***: Enhancing customer satisfaction through accurate order fulfillment and timely delivery.

5. ***Product Tracking***: Keeping track of all products in the inventory, including their location, status, and movement history.
6. ***Reporting and Analytics***: Providing insights into inventory levels, sales trends, and supplier performance to inform business decisions.

Entities for the ERD

For an Inventory Management system, the following entities are essential:

1. ***Product***: Contains details about each product, such as product ID, name, description, price, and SKU (Stock Keeping Unit).
2. ***Stock***: Tracks the quantity of each product available in the inventory, including information on reserved and available stock.
3. ***Supplier***: Holds information about suppliers, including supplier ID, name, contact information, and terms.
4. ***Order***: Represents customer orders, containing order ID, customer ID, order date, and status.
5. ***OrderDetails***: Links products to orders, specifying quantities, prices, and product IDs for each item in an order.
6. ***Customer***: Stores customer information, such as customer ID, name, contact details, and shipping address.
7. ***Employee***: Contains employee details who manage inventory, process orders, or interact with customers.
8. ***Category***: Organizes products into categories or subcategories for easier management and reporting.
9. ***Warehouse***: Represents physical locations where inventory is stored, including warehouse ID, location, and capacity.
10. ***Shipment***: Tracks the shipping details of orders, including shipment ID, carrier, tracking number, and status.
11. ***Payment***: Manages payment information for orders, including payment ID, method, transaction details, and status.

Key Design Decisions

1. ***Normalization***: Entities are designed to minimize redundancy and ensure data integrity. For instance, separating Order and OrderDetails helps manage orders with multiple products more efficiently.

2. *Relationships*:

- *Product to Category*: A many-to-one relationship, as each product belongs to one category, while each category can include many products.
- *Order to Customer*: A many-to-one relationship, where each order is placed by one customer, but each customer can have multiple orders.
- *Order to OrderDetails*: A one-to-many relationship, as each order can contain multiple products (OrderDetails), but each OrderDetail is associated with only one order.
- *Product to Supplier*: Many-to-many, realized through a junction entity, because a product can come from multiple suppliers, and each supplier can provide multiple products.

3. *Use of Foreign Keys*: Foreign keys are used extensively to link related entities, ensuring referential integrity. For example, OrderDetails uses the foreign keys OrderID and ProductID to link to Order and Product, respectively.

4. *Indexes*: Indexes are used on frequently searched fields like ProductID, OrderID, and CustomerID to speed up query performance.

This approach to designing an ERD and the accompanying database design document for an Inventory Management system addresses critical business problems while ensuring data integrity, efficiency, and scalability of the database system.