**Facility Management System**

The best practice for managing materials supplied by suppliers, particularly in terms of tracking them in a warehouse management system, depends on several factors such as the type of materials, the need for traceability, the complexity of inventory management, and the business requirements. Here are some considerations for each approach:

**Best Practice Recommendation**

A **hybrid approach** often provides a balanced solution, combining the strengths of both methods:

1. **Track High-Value or Critical Items Individually**: Use individual serial numbers for items that require detailed tracking and traceability. These could be high-value assets, critical components, or items with warranty or regulatory requirements.
2. **Track Low-Value or Bulk Items by Quantity**: For consumables or low-value items, track them as whole quantities. Register them individually only when they are transferred to users if individual tracking is necessary at that stage.

### Hybrid Approach Implementation

#### Warehouse Table

* **warehouse\_inventory**: Stores bulk quantities and basic information.

CREATE TABLE warehouse\_inventory (

inventory\_id SERIAL PRIMARY KEY,

item\_name VARCHAR(255) NOT NULL,

supplier\_id INTEGER REFERENCES suppliers(supplier\_id),

quantity INTEGER NOT NULL,

received\_date DATE,

location VARCHAR(255)

);

**warehouse\_serialized\_items**: Stores individual items with serial numbers.

CREATE TABLE warehouse\_serialized\_items (

item\_id SERIAL PRIMARY KEY,

inventory\_id INTEGER REFERENCES warehouse\_inventory(inventory\_id),

serial\_number VARCHAR(255) UNIQUE,

status VARCHAR(50),

received\_date DATE,

location VARCHAR(255)

);

#### User Transfer Table

* **user\_transfer**: Records transfer details to users

CREATE TABLE user\_transfer (

transfer\_id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES users(user\_id),

item\_id INTEGER REFERENCES warehouse\_serialized\_items(item\_id),

transfer\_date DATE

);

**Workflow**

1. **Receiving Inventory**:
   * Bulk items are recorded in warehouse\_inventory with quantity.
   * Serialized items are recorded in warehouse\_serialized\_items linked to warehouse\_inventory.
2. **Transferring to Users**:
   * When bulk items are transferred, update the quantity in warehouse\_inventory.
   * When serialized items are transferred, create entries in user\_transfer linking to warehouse\_serialized\_items.

**Possible Table In the project**

Public

public class Disposal

{

public int disposalId { get; set; }

public int itemName { get; set; }

public string quantity{ get; set; }

public DateTime disposal\_Date { get; set; }

public string disposal\_Method { get; set; }

public string disposalReason { get; set; }

public int disposalCost { get; set; }

public int disposedBy { get; set; }//Foriegn key the Id of the User Table

public int approvedBy { get; set; }//Foriegn key for the User table

public string location { get; set; }

}

public class IssuedMaterials

{

public int Issuance\_ID { get; set; }

public string materialName { get; set; }

public string serialNumber{ get; set; }

public int toUser { get; set; }//Foriegn key for User table

public string condition { get; set; }//Condition: The condition of the material being issued (e.g., new, used).

public int approvedBy { get; set; }//Foriegn Key from User Table

public DateTime issueDate { get; set; }

}

public class MaterialReturn

{

public int returnId { get; set; }

public int userId { get; set; }//Foriegn Key from the user Id

public string materialName{ get; set; }

public int workingUnitId { get; set; }//Foriegn key for Working Unit Id

public int returnedQuantity { get; set; }

public DateTime returnDate { get; set; }

public string returnPurpose { get; set; }

public string condition { get; set; }//Condition: The condition of the material being returned (e.g., new, used, damaged).

public int approvedBy { get; set; }//Foriegn key for the User table

public DateTime approvalDate { get; set; }

///there may be another attributes

}

public class MaterialTransfer

{

public int transferId { get; set; }

public int fromUser { get; set; }//Foriegn Key for a user table

public int toUser { get; set; }//Foriegn key for a user table

public int materialId { get; set; }//Forign key for IssuedProperty table

public string condition { get; set; }//Condition: The condition of the material being transferred (e.g., new, used, damaged).

public int approvedBy { get; set; }//foriegn key for user table there may be other attributes later!

}

public class PurchaseOrder

{

public int purchase\_Order\_Number { get; set; }

public string supplierId{ get; set; }

// This will be a Foriegn key

public string material { get; set; }

public int quantity { get; set; }

public int unit\_Price { get; set; }

public int totalPrice { get; set; }

public Date orderDate { get; set; }

// There may others like order Delivering Date,and Others

}

public class Supplier

{

public int supplierId { get; set; }

public string supplier\_Name { get; set; }

public string contact\_Person { get; set; }

public int phone\_Number { get; set; }

public string email\_Address { get; set; }

public int taxId { get; set; }

public string bank\_Name { get; set; }

public int account\_Number { get; set; }

public string swiftcode { get; set; }

public string swift\_Bic\_Code { get; set; }

public string supplier\_Type { get; set; }

public string category { get; set; }

}

public class User

{

public int Id { get; set; }

public string First\_Name { get; set; }

public string Last\_Name { get; set; }

public int workingUnitId { get; set; }//Foriegn Key Which this User Belongs to

}

public class UserRequest

{

public int requestId { get; set; }

public int userId { get; set; }//Foriegn Key for the User Table

public int workingUnitId { get; set; }//Foriegn key for Working Unit table

public string materialName { get; set; }

public int requestedQuantity { get; set; }

public DateTime requestedDate { get; set; }

public string purpose { get; set; }

public string approvalStatus { get; set; }

public string approvedBy{ get; set; }

public DateTime approvedDate { get; set; }

// Request\_ID: A unique identifier for each material request.

//User\_ID: The unique identifier of the user making the request.

//User\_Name: The name of the user making the request.

//Department: The department to which the user belongs.

//Material\_ID: The unique identifier for the requested material.

//Material\_Name: The name of the requested material.

//Quantity\_Requested: The quantity of the material requested.

//Request\_Date: The date when the request was made.

//Priority\_Level: The priority level of the request (e.g., low, medium, high).

//Purpose: The reason or purpose for requesting the material.

//Approval\_Status: The current status of the request (e.g., pending, approved, rejected).

//Approved\_By: The name or ID of the person who approved the request.

//Approval\_Date: The date when the request was approved.

}

public class warehouse\_Serialized\_Items

{

public int Id { get; set; }

public int inventoryId { get; set; }//Has Foriegn Key with wareHouseInventory

public string serial\_Number { get; set; }

// item\_id SERIAL PRIMARY KEY,

//inventory\_id INTEGER REFERENCES warehouse\_inventory(inventory\_id),

// serial\_number VARCHAR(255) UNIQUE,

// status VARCHAR(50),

// received\_date DATE,

// location VARCHAR(255)

}

public class wareHouseInventory

{

public int InventoryId { get; set; }

public string item\_Name { get; set; }

public int supplierId { get; set; }//This will the Supplier as a Foriegn key

public string quantity { get; set; }

public DateTime recieved\_Date { get; set; }

public string location { get; set; }

}