

```
-- =====  
-- Warehouse Management System (WMS)  
-- Struktur Database  
-- =====
```

```
-- Master Data
```

```
CREATE TABLE roles (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(150) NOT NULL,  
    email VARCHAR(150) UNIQUE NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    role_id INT,  
    FOREIGN KEY (role_id) REFERENCES roles(id)  
);
```

```
CREATE TABLE warehouses (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(150) NOT NULL,  
    code VARCHAR(50) UNIQUE NOT NULL  
);
```

```
CREATE TABLE locations (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    warehouse_id INT NOT NULL,  
    code VARCHAR(50) NOT NULL,  
    description VARCHAR(150),
```

```
FOREIGN KEY (warehouse_id) REFERENCES warehouses(id)
);
```

```
CREATE TABLE product_categories (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL
);
```

```
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    sku VARCHAR(100) UNIQUE NOT NULL,
    name VARCHAR(150) NOT NULL,
    category_id INT,
    unit VARCHAR(50),
    FOREIGN KEY (category_id) REFERENCES product_categories(id)
);
```

```
CREATE TABLE suppliers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    contact VARCHAR(150),
    phone VARCHAR(50)
);
```

```
CREATE TABLE customers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    contact VARCHAR(150),
    phone VARCHAR(50)
);
```

-- Inbound (Purchase Order & Receiving)

```
CREATE TABLE purchase_orders (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    supplier_id INT NOT NULL,  
    warehouse_id INT NOT NULL,  
    order_date DATE NOT NULL,  
    status ENUM('draft','approved','received','cancelled') DEFAULT 'draft',  
    FOREIGN KEY (supplier_id) REFERENCES suppliers(id),  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id)  
);
```

```
CREATE TABLE purchase_order_items (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    purchase_order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    qty_ordered INT NOT NULL,  
    FOREIGN KEY (purchase_order_id) REFERENCES purchase_orders(id),  
    FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

```
CREATE TABLE goods_receipts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    purchase_order_id INT,  
    receipt_date DATE NOT NULL,  
    received_by INT,  
    FOREIGN KEY (purchase_order_id) REFERENCES purchase_orders(id),  
    FOREIGN KEY (received_by) REFERENCES users(id)  
);
```

```
CREATE TABLE goods_receipt_items (  
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
goods_receipt_id INT NOT NULL,  
product_id INT NOT NULL,  
qty_received INT NOT NULL,  
FOREIGN KEY (goods_receipt_id) REFERENCES goods_receipts(id),  
FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

-- Inventory

```
CREATE TABLE stocks (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    warehouse_id INT NOT NULL,  
    product_id INT NOT NULL,  
    qty INT NOT NULL DEFAULT 0,  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id),  
    FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

```
CREATE TABLE stock_movements (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    product_id INT NOT NULL,  
    warehouse_id INT NOT NULL,  
    movement_type ENUM('in','out') NOT NULL,  
    qty INT NOT NULL,  
    reference_table VARCHAR(100),  
    reference_id INT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (product_id) REFERENCES products(id),  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id)  
);
```

```
CREATE TABLE stock_adjustments (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    product_id INT NOT NULL,  
    warehouse_id INT NOT NULL,  
    movement_type ENUM('in','out') NOT NULL,  
    qty INT NOT NULL,  
    reference_table VARCHAR(100),  
    reference_id INT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (product_id) REFERENCES products(id),  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id)  
);
```

```
id INT AUTO_INCREMENT PRIMARY KEY,
product_id INT NOT NULL,
warehouse_id INT NOT NULL,
qty_before INT NOT NULL,
qty_after INT NOT NULL,
reason VARCHAR(255),
adjusted_by INT,
adjusted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (product_id) REFERENCES products(id),
FOREIGN KEY (warehouse_id) REFERENCES warehouses(id),
FOREIGN KEY (adjusted_by) REFERENCES users(id)
);
```

-- Outbound (Sales Order & Shipment)

```
CREATE TABLE sales_orders (
id INT AUTO_INCREMENT PRIMARY KEY,
customer_id INT NOT NULL,
warehouse_id INT NOT NULL,
order_date DATE NOT NULL,
status ENUM('draft','confirmed','shipped','cancelled') DEFAULT 'draft',
FOREIGN KEY (customer_id) REFERENCES customers(id),
FOREIGN KEY (warehouse_id) REFERENCES warehouses(id)
);
```

```
CREATE TABLE sales_order_items (
id INT AUTO_INCREMENT PRIMARY KEY,
sales_order_id INT NOT NULL,
product_id INT NOT NULL,
qty_ordered INT NOT NULL,
FOREIGN KEY (sales_order_id) REFERENCES sales_orders(id),
FOREIGN KEY (product_id) REFERENCES products(id)
```

);

```
CREATE TABLE pick_lists (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    sales_order_id INT NOT NULL,  
    picker_id INT,  
    pick_date DATE,  
    FOREIGN KEY (sales_order_id) REFERENCES sales_orders(id),  
    FOREIGN KEY (picker_id) REFERENCES users(id)  
);
```

```
CREATE TABLE pick_list_items (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    pick_list_id INT NOT NULL,  
    product_id INT NOT NULL,  
    qty_picked INT NOT NULL,  
    FOREIGN KEY (pick_list_id) REFERENCES pick_lists(id),  
    FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

```
CREATE TABLE shipments (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    sales_order_id INT NOT NULL,  
    shipped_date DATE,  
    courier VARCHAR(150),  
    tracking_number VARCHAR(150),  
    shipped_by INT,  
    FOREIGN KEY (sales_order_id) REFERENCES sales_orders(id),  
    FOREIGN KEY (shipped_by) REFERENCES users(id)  
);
```

-- Returns

```
CREATE TABLE returns (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    type ENUM('sales_return','purchase_return') NOT NULL,  
    reference_id INT,  
    return_date DATE,  
    status ENUM('pending','processed','cancelled') DEFAULT 'pending'  
);
```

```
CREATE TABLE return_items (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    return_id INT NOT NULL,  
    product_id INT NOT NULL,  
    qty INT NOT NULL,  
    FOREIGN KEY (return_id) REFERENCES returns(id),  
    FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

-- Audit Logs

```
CREATE TABLE audit_logs (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    action VARCHAR(255),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES users(id)  
);
```