

لإنشاء قاعدة بيانات شاملة لنظام إدارة المخازن الذي يشمل الحركات المختلفة (إضافة، صرف، مرتجع داخلي، مرتجع للمورد، وإهلاك)، يمكنك استخدام الجداول التالية:

تصميم قاعدة البيانات

جدول الأصناف (Items)

يحتوي على تفاصيل الأصناف المخزنة.

```
sql
Copy code
CREATE TABLE Items (
    item_id INT PRIMARY KEY AUTO_INCREMENT,
    item_code VARCHAR(20) NOT NULL,
    item_name VARCHAR(100) NOT NULL,
    item_type VARCHAR(50),
    unit VARCHAR(20),
    reorder_level INT
);
```

جدول المخازن (Warehouses)

يحتوي على تفاصيل المخازن المتاحة.

```
sql
Copy code
CREATE TABLE Warehouses (
    warehouse_id INT PRIMARY KEY AUTO_INCREMENT,
    warehouse_name VARCHAR(100) NOT NULL
);
```

جدول المخزون (Inventory)

يحتوي على الكميات المخزنة لكل صنف في كل مخزن.

```
sql
Copy code
CREATE TABLE Inventory (
    inventory_id INT PRIMARY KEY AUTO_INCREMENT,
    warehouse_id INT,
    item_id INT,
    quantity INT,
    FOREIGN KEY (warehouse_id) REFERENCES Warehouses(warehouse_id),
    FOREIGN KEY (item_id) REFERENCES Items(item_id)
);
```

جدول الأسباب (Reasons)

يحتوي على الأسباب المرتبطة بالحركات المختلفة.

```
sql
Copy code
CREATE TABLE Reasons (
    reason_id INT PRIMARY KEY AUTO_INCREMENT,
    reason_description VARCHAR(255) NOT NULL
);
```

جدول الحركات (Transactions)

يحتوي على تفاصيل الحركات المخزنية بما في ذلك المرتجعات والإهلاك

sql

Copy code

```
CREATE TABLE Transactions (  
    transaction_id INT PRIMARY KEY AUTO_INCREMENT,  
    transaction_date DATE NOT NULL,  
    warehouse_id INT,  
    item_id INT,  
    quantity INT,  
    transaction_type VARCHAR(20), -- يمكن أن تكون (إضافة، صرف، مرتجع داخلي، مرتجع  
    للمورد، إهلاك)  
    recipient VARCHAR(100), -- جهة الصرف  
    return_type VARCHAR(20), -- يمكن أن تكون (داخلي، للمورد)  
    return_code VARCHAR(20),  
    depreciation_code VARCHAR(20),  
    reason_id INT,  
    document_number VARCHAR(20),  
    FOREIGN KEY (warehouse_id) REFERENCES Warehouses(warehouse_id),  
    FOREIGN KEY (item_id) REFERENCES Items(item_id),  
    FOREIGN KEY (reason_id) REFERENCES Reasons(reason_id)  
);
```

جدول الإهلاك (Depreciations)

يحتوي على تفاصيل إذن الإهلاك

sql

Copy code

```
CREATE TABLE Depreciations (  
    depreciation_id INT PRIMARY KEY AUTO_INCREMENT,  
    item_id INT,  
    warehouse_id INT,  
    quantity INT,  
    depreciation_code VARCHAR(20),  
    reason_id INT,  
    depreciation_date DATE,  
    FOREIGN KEY (item_id) REFERENCES Items(item_id),  
    FOREIGN KEY (warehouse_id) REFERENCES Warehouses(warehouse_id),  
    FOREIGN KEY (reason_id) REFERENCES Reasons(reason_id)  
);
```

العلاقات بين الجداول:

- **item_id** باستخدام **Depreciations** و **Transactions** و **Inventory** ترتبط بـ **Items**.
- **warehouse_id** باستخدام **Depreciations** و **Transactions** و **Inventory** ترتبط بـ **Warehouses**.
- **reason_id** باستخدام **Reasons** ترتبط بـ **Transactions**.
- **reason_id** و **item_id** و **warehouse_id** باستخدام **Reasons** و **Items** و **Warehouses** ترتبط بـ **Depreciations**.

بهذا التصميم، يمكنك إدارة جميع الحركات المخزنية المطلوبة بما في ذلك الإضافات، الصرف، المرتجعات، والإهلاك، مع توفير الأسباب المرتبطة بكل حركة.