

# **Building a Custom Sales Blocklist in**



**Business One**

**Amirhossein Tonekaboni**

**SAP Business One ERP Consultant**



[Linkedin.com/in/Tonekaboni](https://www.linkedin.com/in/Tonekaboni)

## Abstract

How can we block specific items for certain customers in SAP Business One? Unchecking the "Saleable" field in the Item Master Data will block the item for all customers, but what if we want to give clients the option to choose which items are not saleable for each customer during a specific period? The solution is to define a Sales Blocklist table in the system and use an SQL stored procedure to prevent sales forms from recording restricted data.

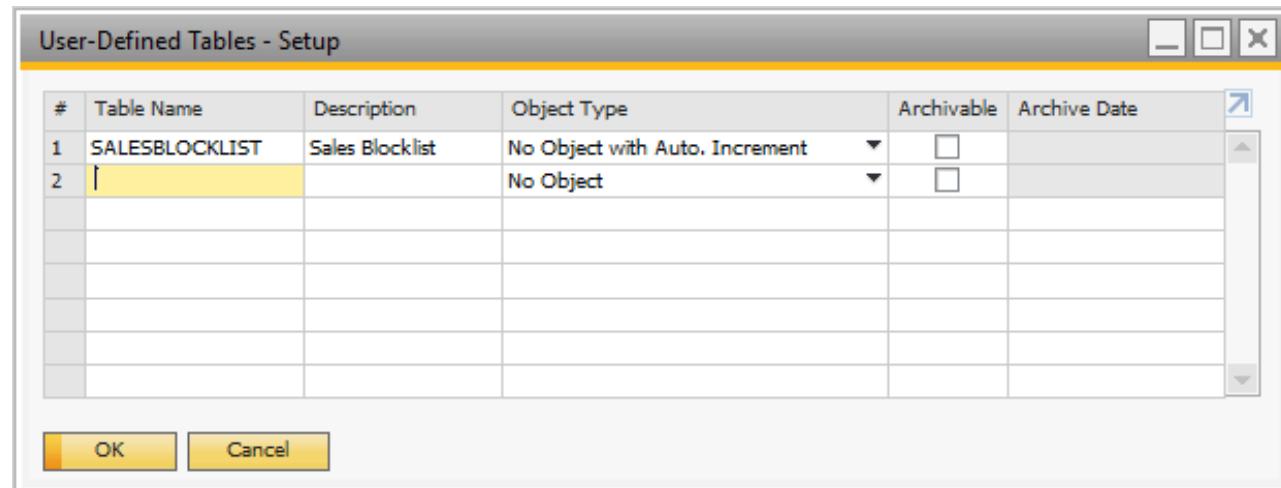
## Steps

1. User-Defined Tables – Setup
  2. User-Defined Fields – Management
  3. Modifying Stored Procedure in the Database
  4. Filling Out the Designated Table
  5. Sales Documents

## 1. User-Defined Tables – Setup

To create tables in SAP Business One, follow this path and create the Sales Blocklist table:

*Tools → Customization Tools → User-Defined Tables – Setup*

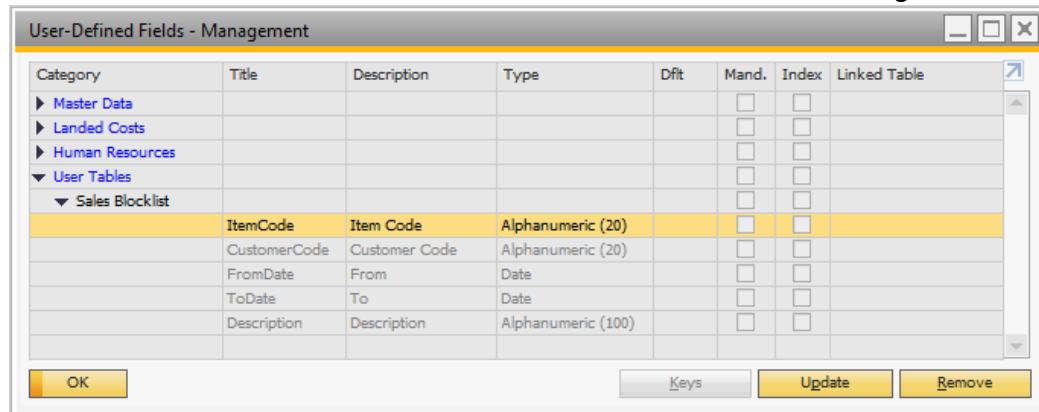


 For an auto-generated numeric key, set **Object Type** to **No Object with Auto Increment**; this automatically fills the Code column. Moreover, there would be no risk of duplicate keys and no need for custom numbering logic.

## 2. User-Defined Fields – Management

To create columns in the User-Defined Fields Management form:

*Tools → Customization Tools → User-Defined Fields – Management*



After the table is saved, SAP Business One adds it to Tools → User-Defined Windows.

## 3. Modifying Stored Procedure in the Database

Modify **SBO\_SP\_TransactionNotification** in the selected database.

```
IF @object_type = '17' AND @transaction_type IN ('A', 'U') -- Sales Order
BEGIN
    DECLARE @CardCode NVARCHAR(50), @DocDate DATE
    DECLARE @BlockedItems NVARCHAR(MAX) = ""

    SELECT @CardCode = CardCode, @DocDate = DocDate FROM ORDR WHERE DocEntry = @list_of_cols_val_tab_del

    SELECT @BlockedItems = STRING_AGG(T0.ItemCode, ', ')
    FROM RDR1 TO
    INNER JOIN [@SALESBLOCKLIST] T1
        ON T1.U_CustomerCode = @CardCode
        AND T1.U_ItemCode = T0.ItemCode
    WHERE T0.DocEntry = @list_of_cols_val_tab_del
        AND @DocDate BETWEEN T1.U_FromDate AND T1.U_ToDate

    IF @BlockedItems IS NOT NULL AND @BlockedItems <> ""
    BEGIN
        SET @error = 1001
        SET @error_message = N'Selling these items to this customer on this date is not allowed: ' + @BlockedItems
    END
END
```

The code above applies the blocklist logic to Sales Orders. For the complete script, which extends this validation to other documents like **Deliveries** and **A/R Invoices**, please see the full version on [GitHub](#).

#### 4. Filling Out the Designated Table

Fill the table manually or by copying data from Excel spreadsheets. Open the created table from the menu and fill it up for the specific restrictions.

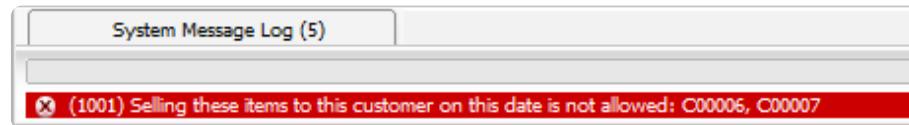
The screenshot shows a 'Sales Blocklist' dialog box and an open Excel spreadsheet. The dialog box contains a table with columns: #, Code, Customer Code, Item Code, From, To, and Description. It lists two entries: row 1 with code C60000, item C00006, and description '3-Months Restriction Period'; and row 2 with code C60000, item C00007, and description '2-Months Restriction Period'. Below the table are 'OK' and 'Cancel' buttons. To the right of the dialog is an Excel window titled 'Book1 - Excel' with a single row of data: A1 contains 'C60000', B1 contains 'C00007', C1 contains '20250501', D1 contains '20250701', and E1 contains '2-Months Restriction Period'. The Excel ribbon shows tabs for File, Home, Insert, Page Layout, Formulas, Data, Review, and View.

#### 5. Sales Documents

In the following step, use a Sales documents like Sales Order by selecting a specific customer and items.

The screenshot shows the 'Sales Order' window. At the top, it displays customer information: Customer C60000, Name SG Electronics, Contact Person Eric Alexander, and BP Currency \$. On the right, it shows document details: No. 1201, Status Open, Posting Date 05/12/2025, Delivery Date 05/12/2025, and Document Date 05/12/2025. Below this is a grid of items with columns: #, Item No., Item Description, Quantity, Discount %, Unit Price, Sales Type, Whse, Del. Date, and Total (LC). The grid contains four items: 1. C00005 WLAN Card, 5 units at \$90.00, total \$450.00. 2. C00006 Gigabit Network Card, 10 units at \$22.50, total \$225.00. 3. C00007 Hard Disk 3TB, 20 units at \$750.00, total \$15,000.00. 4. (empty). At the bottom, it shows sales employee Jim Boswick, remarks, and financial summary: Total Before Discount 15,675.00 \$, Discount 0%, Freight 0.00 \$, Tax 0.00 \$, and Total 15,675.00 \$. Buttons at the bottom include 'Add & View', 'Cancel', 'Copy From', and 'Copy To'.

Now, when we select items for the specific customer that is on the Sales Blocklist table, we will see an error that forbids the user from selecting these items.



## Conclusion

This method is a useful way to **restrict** the sale of **specific items** to **certain customers** for a **defined period of time** in SAP Business One. For the blocking logic itself, several alternatives can also be very effective. These include using the B1 Usability Package (a third-party add-on) or adding a User-Defined Field (UDF) at the row level with a Formatted Search for real-time validation. Additionally, for managing large volumes of restriction data, using the Data Transfer Workbench (DTW) for bulk uploads can be a useful tool for larger implementations.

## Prepared by

**Amirhossein Tonekaboni**  
**SAP Business One ERP Consultant**

-  [GitHub](#)
-  [Linkedin](#)
-  [Allmylinks](#)