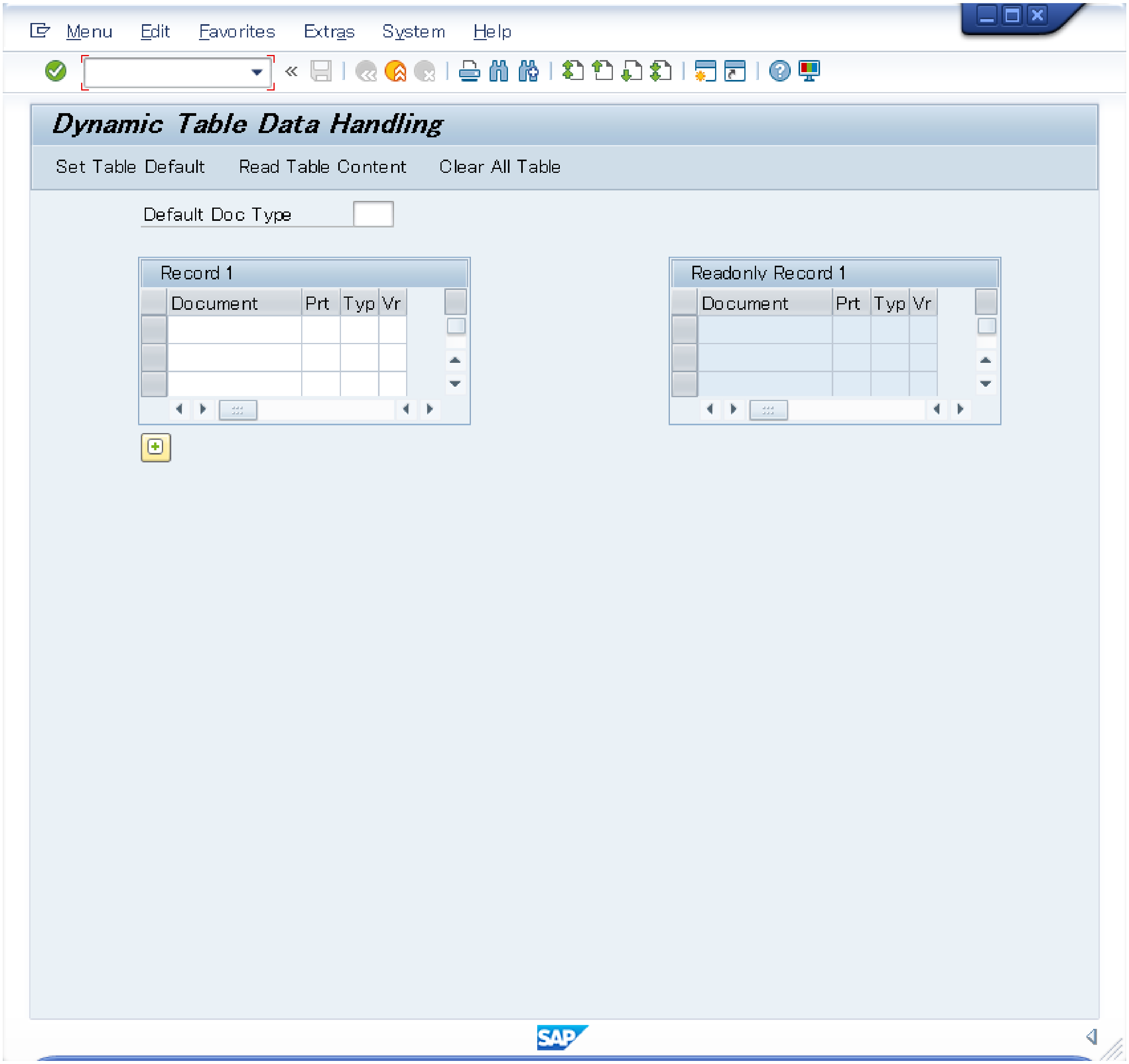
Liquid UI: Dynamic create table, and read / write value into table cell

User Interface



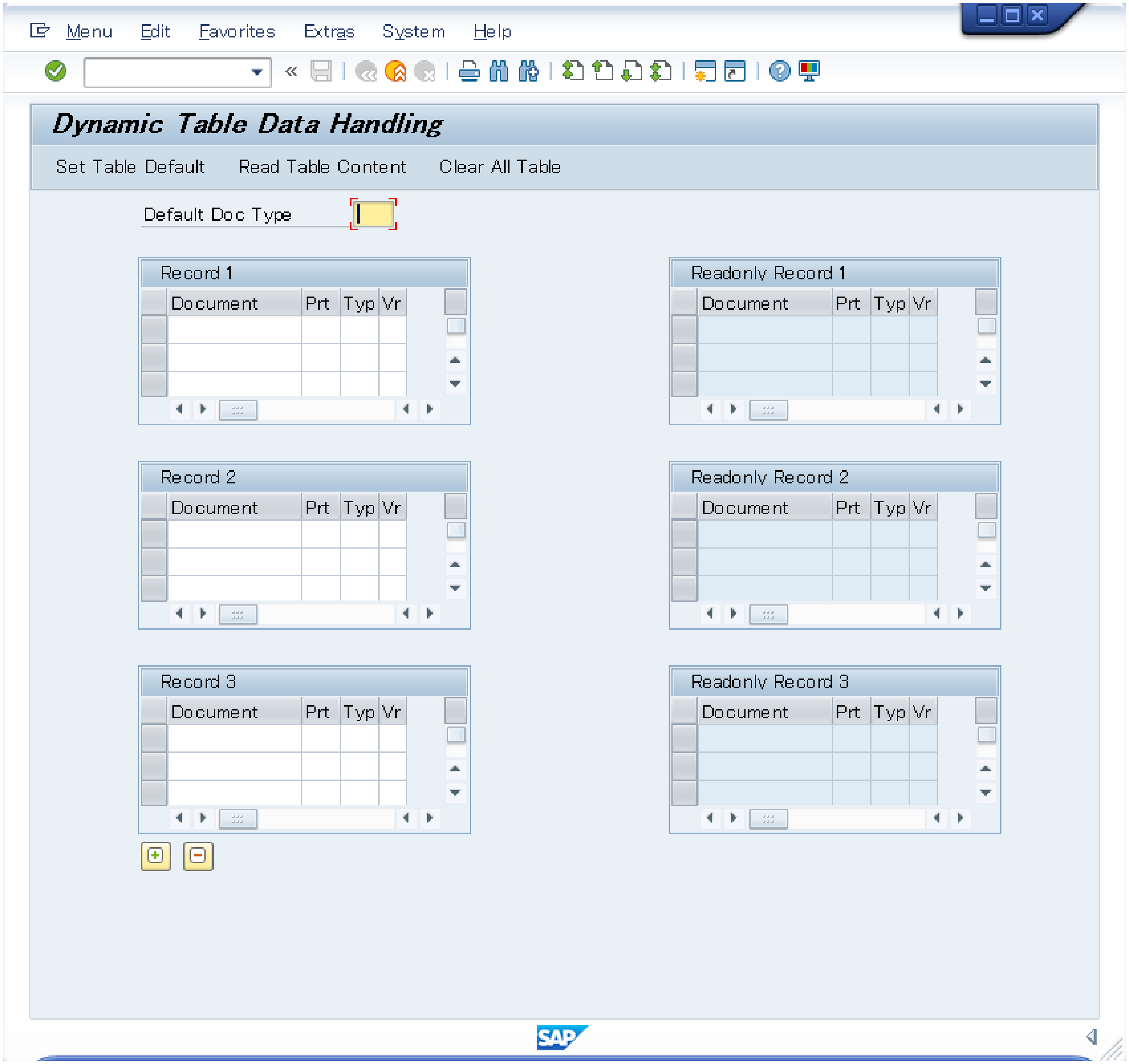
Record 1

Document	Prt	Typ	Vr

Readonly Record 1

Document	Prt	Typ	Vr

After add two new tables



Default Doc Type

Record 1

Document	Prt	Typ	Vr

Readonly Record 1

Document	Prt	Typ	Vr

Record 2

Document	Prt	Typ	Vr

Readonly Record 2

Document	Prt	Typ	Vr

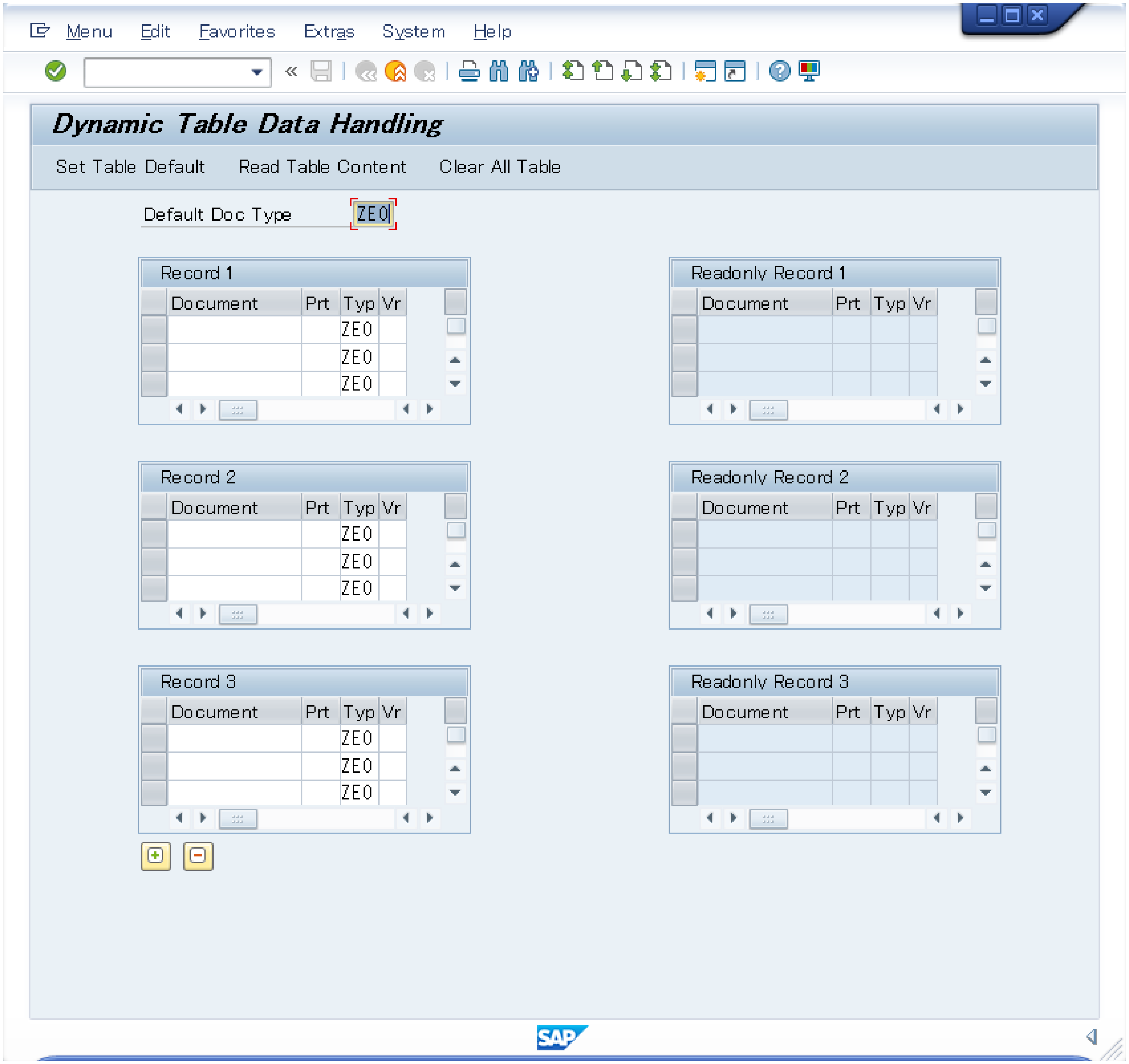
Record 3

Document	Prt	Typ	Vr

Readonly Record 3

Document	Prt	Typ	Vr

After click “Set Table Default” to set entered Doc type to all editable tables



Dynamic Table Data Handling

Default Doc Type ZE0

Record 1

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 1

Document	Prt	Typ	Vr

Record 2

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 2

Document	Prt	Typ	Vr

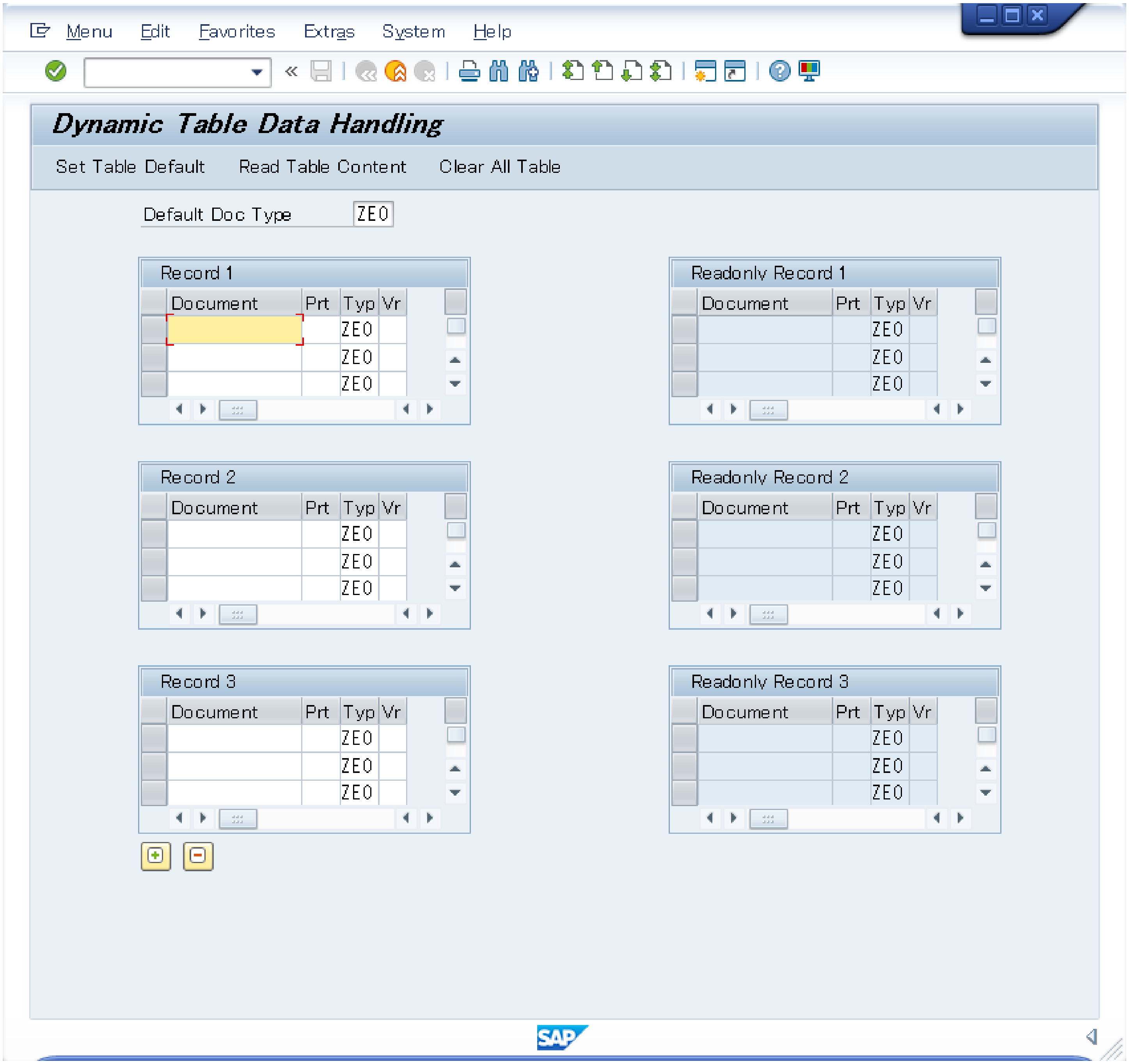
Record 3

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 3

Document	Prt	Typ	Vr

After click “Read Table Content” to copy value from editable tables to readonly tables



Dynamic Table Data Handling

Default Doc Type ZE0

Record 1

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 1

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Record 2

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 2

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

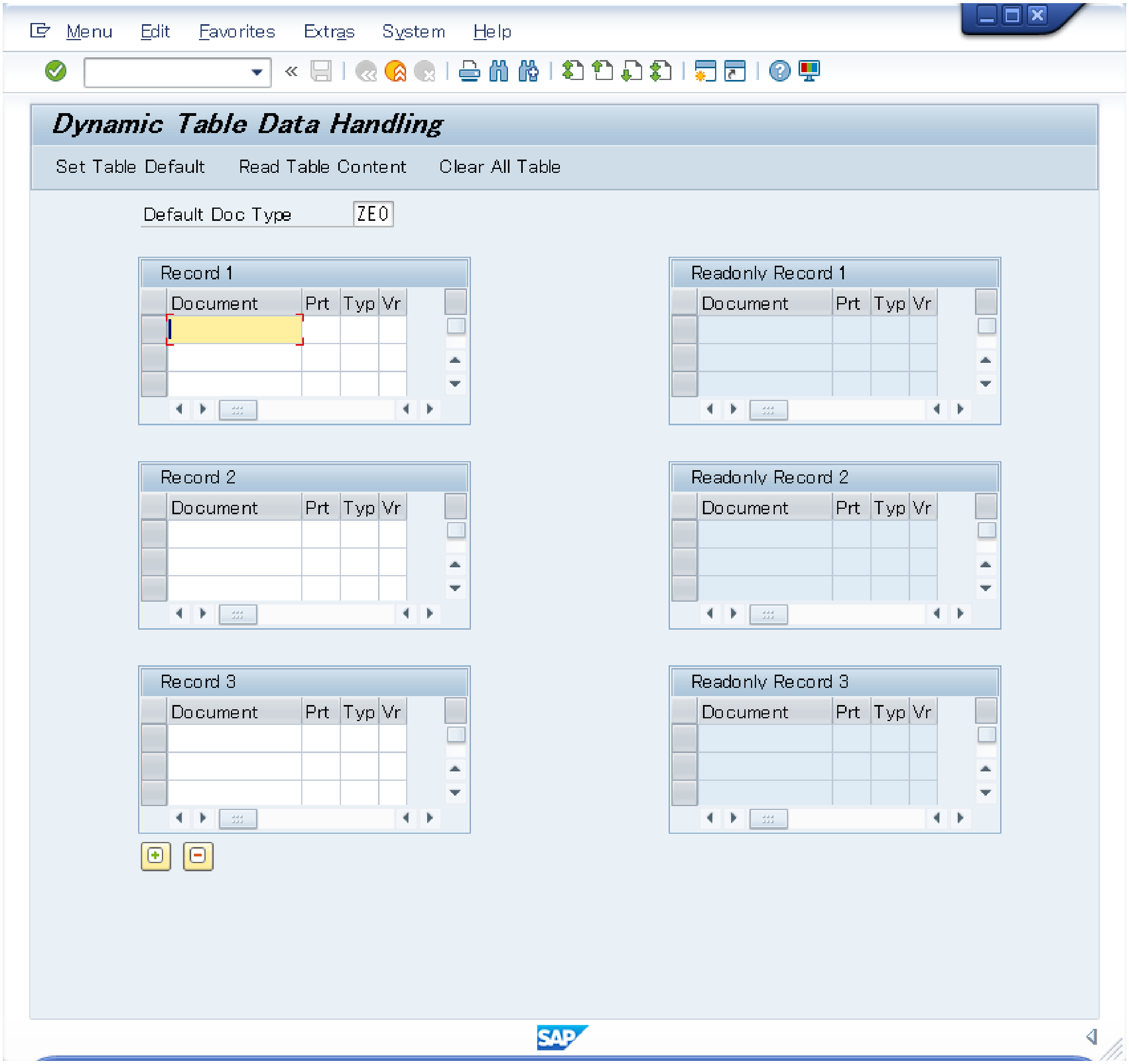
Record 3

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

Readonly Record 3

Document	Prt	Typ	Vr
		ZE0	
		ZE0	
		ZE0	

After click “Clear All Table”



Default Doc Type ZE0

Record 1

Document	Prt	Typ	Vr

Readonly Record 1

Document	Prt	Typ	Vr

Record 2

Document	Prt	Typ	Vr

Readonly Record 2

Document	Prt	Typ	Vr

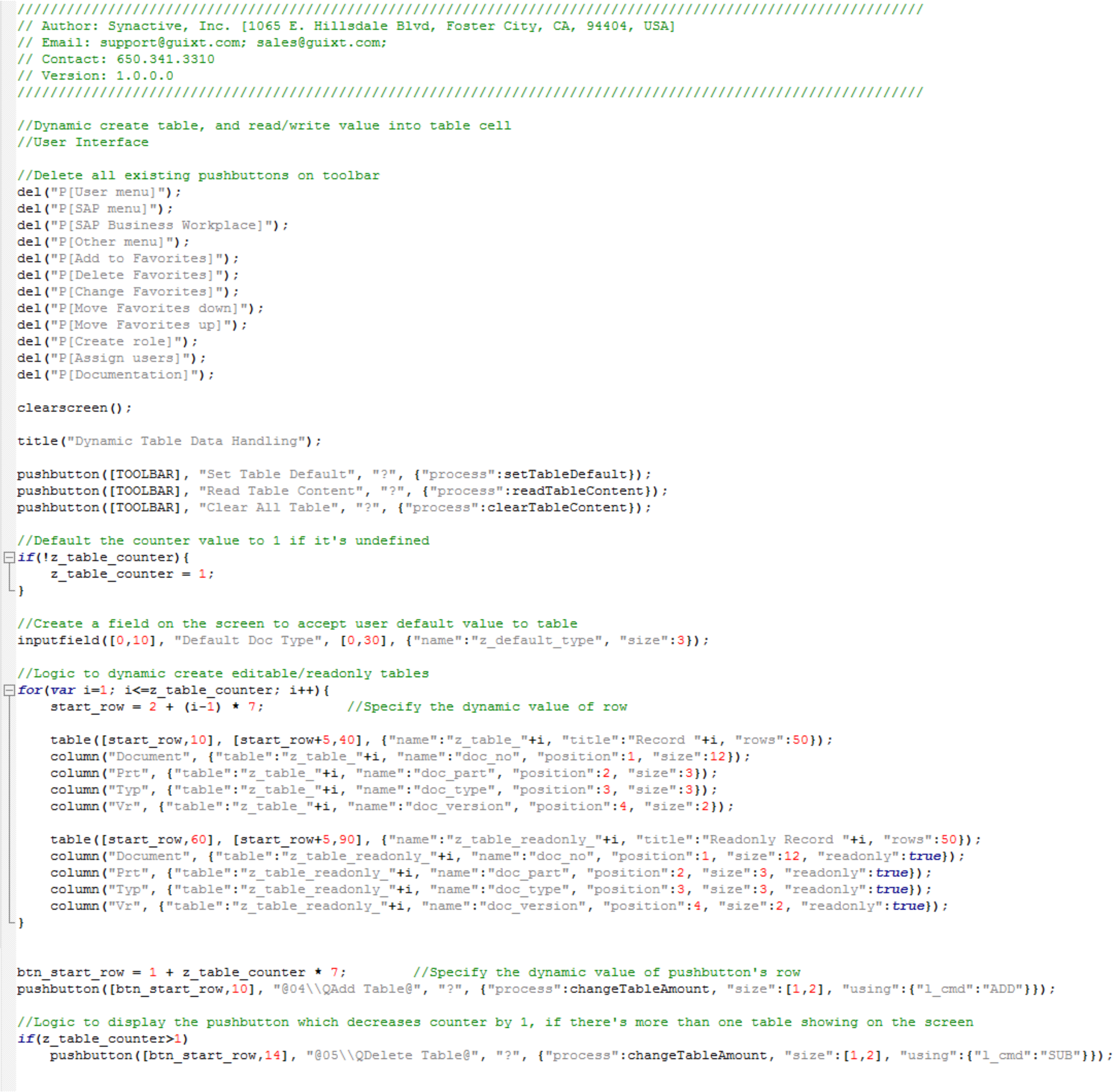
Record 3

Document	Prt	Typ	Vr

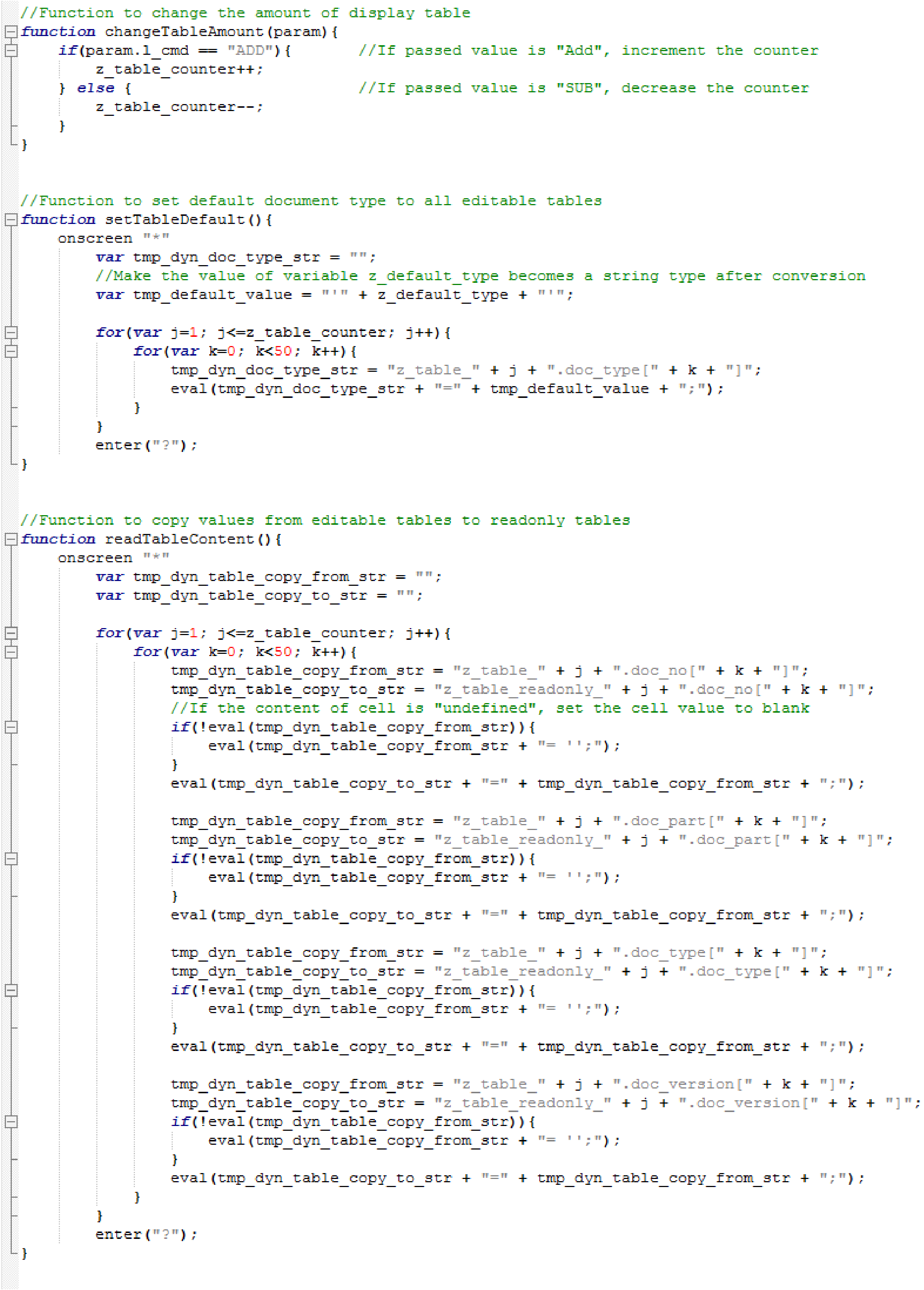
Readonly Record 3

Document	Prt	Typ	Vr

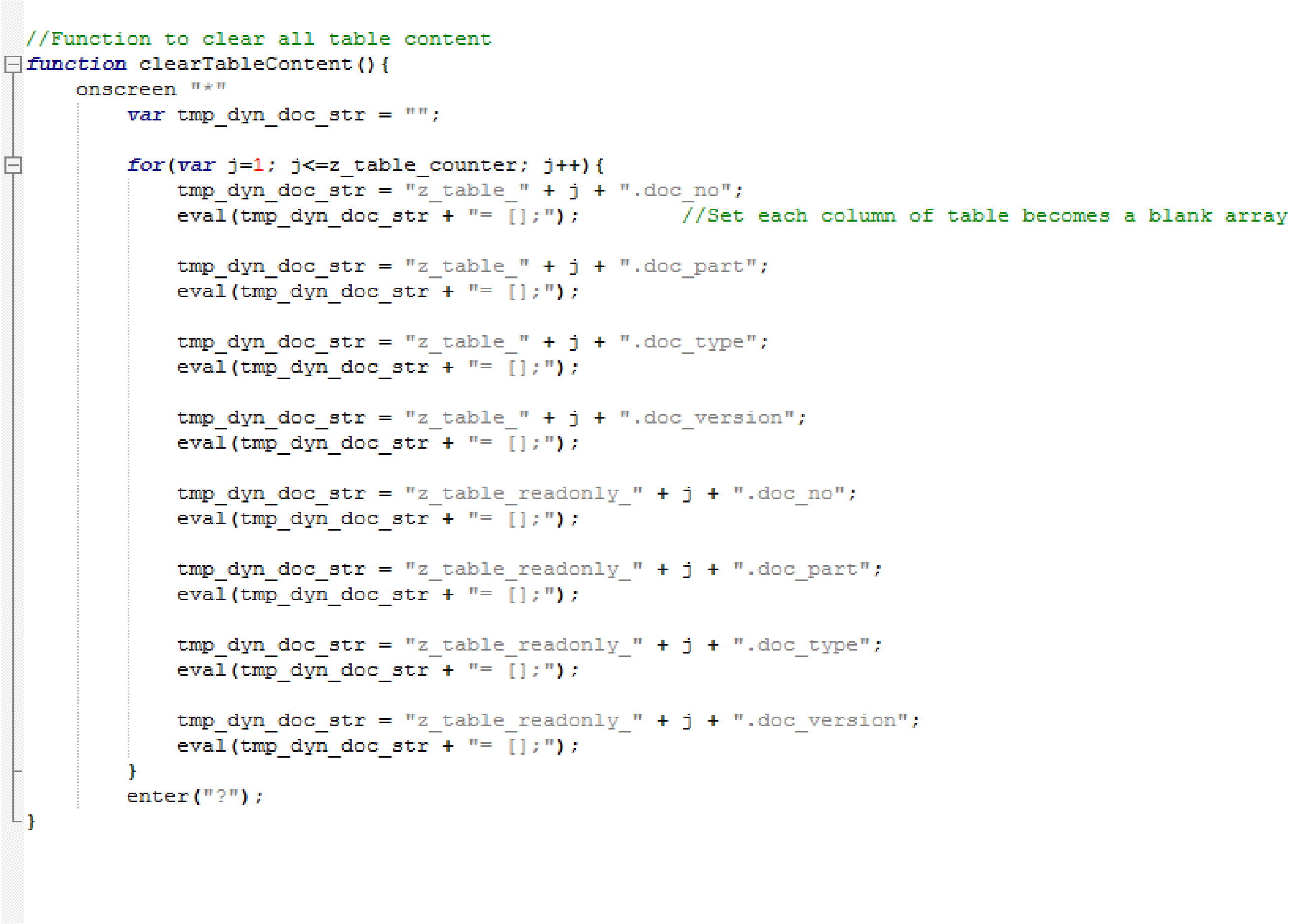
Liquid UI Code [Script]



```
////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Author: Synactive, Inc. [1065 E. Hillsdale Blvd, Foster City, CA, 94404, USA]
// Email: support@guixt.com; sales@guixt.com;
// Contact: 650.341.3310
// Version: 1.0.0.0
////////////////////////////////////////////////////////////////////////////////////////////////////////////

//Dynamic create table, and read/write value into table cell
//User Interface

//Delete all existing pushbuttons on toolbar
del("P[User menu]");
del("P[SAP menu]");
del("P[SAP Business Workplace]");
del("P[Other menu]");
del("P[Add to Favorites]");
del("P[Delete Favorites]");
del("P[Change Favorites]");
del("P[Move Favorites down]");
del("P[Move Favorites up]");
del("P[Create role]");
del("P[Assign users]");
del("P[Documentation]");

clearscreen();

title("Dynamic Table Data Handling");

pushbutton([TOOLBAR], "Set Table Default", "?", {"process":setTableDefault});
pushbutton([TOOLBAR], "Read Table Content", "?", {"process":readTableContent});
pushbutton([TOOLBAR], "Clear All Table", "?", {"process":clearTableContent});

//Default the counter value to 1 if it's undefined
if(!z_table_counter){
    z_table_counter = 1;
}

//Create a field on the screen to accept user default value to table
inputfield([0,10], "Default Doc Type", [0,30], {"name":"z_default_type", "size":3});

//Logic to dynamic create editable/readonly tables
for(var i=1; i<=z_table_counter; i++){
    start_row = 2 + (i-1) * 7;          //Specify the dynamic value of row

    table([start_row,10], [start_row+5,40], {"name":"z_table_"+i, "title":"Record "+i, "rows":50});
    column("Document", {"table":"z_table_"+i, "name":"doc_no", "position":1, "size":12});
    column("Prt", {"table":"z_table_"+i, "name":"doc_part", "position":2, "size":3});
    column("Typ", {"table":"z_table_"+i, "name":"doc_type", "position":3, "size":3});
    column("Vr", {"table":"z_table_"+i, "name":"doc_version", "position":4, "size":2});

    table([start_row,60], [start_row+5,90], {"name":"z_table_readonly_"+i, "title":"Readonly Record "+i, "rows":50});
    column("Document", {"table":"z_table_readonly_"+i, "name":"doc_no", "position":1, "size":12, "readonly":true});
    column("Prt", {"table":"z_table_readonly_"+i, "name":"doc_part", "position":2, "size":3, "readonly":true});
    column("Typ", {"table":"z_table_readonly_"+i, "name":"doc_type", "position":3, "size":3, "readonly":true});
    column("Vr", {"table":"z_table_readonly_"+i, "name":"doc_version", "position":4, "size":2, "readonly":true});
}


btn_start_row = 1 + z_table_counter * 7;         //Specify the dynamic value of pushbutton's row
pushbutton([btn_start_row,10], "@04\\QAdd Table@", "?", {"process":changeTableAmount, "size":[1,2], "using":{"l_cmd":"ADD"}});

//Logic to display the pushbutton which decreases counter by 1, if there's more than one table showing on the screen
if(z_table_counter>1)
    pushbutton([btn_start_row,14], "@05\\QDelete Table@", "?", {"process":changeTableAmount, "size":[1,2], "using":{"l_cmd":"SUB"}});
```



```
//Function to change the amount of display table
function changeTableAmount(param){
    if(param.l_cmd == "ADD"){       //If passed value is "Add", increment the counter
        z_table_counter++;
    } else {                        //If passed value is "SUB", decrease the counter
        z_table_counter--;
    }
}


//Function to set default document type to all editable tables
function setTableDefault(){
    onscreen "*"
        var tmp_dyn_doc_type_str = "";
        //Make the value of variable z_default_type becomes a string type after conversion
        var tmp_default_value = "'" + z_default_type + "'";

        for(var j=1; j<=z_table_counter; j++){
            for(var k=0; k<50; k++){
                tmp_dyn_doc_type_str = "z_table_" + j + ".doc_type[" + k + "]";
                eval(tmp_dyn_doc_type_str + "=" + tmp_default_value + ";");
            }
        }
        enter("?");
}


//Function to copy values from editable tables to readonly tables
function readTableContent(){
    onscreen "*"
        var tmp_dyn_table_copy_from_str = "";
        var tmp_dyn_table_copy_to_str = "";

        for(var j=1; j<=z_table_counter; j++){
            for(var k=0; k<50; k++){
                tmp_dyn_table_copy_from_str = "z_table_" + j + ".doc_no[" + k + "]";
                tmp_dyn_table_copy_to_str = "z_table_readonly_" + j + ".doc_no[" + k + "]";
                //If the content of cell is "undefined", set the cell value to blank
                if(!eval(tmp_dyn_table_copy_from_str)){
                    eval(tmp_dyn_table_copy_from_str + "= '';");
                }
                eval(tmp_dyn_table_copy_to_str + "=" + tmp_dyn_table_copy_from_str + ";");

                tmp_dyn_table_copy_from_str = "z_table_" + j + ".doc_part[" + k + "]";
                tmp_dyn_table_copy_to_str = "z_table_readonly_" + j + ".doc_part[" + k + "]";
                if(!eval(tmp_dyn_table_copy_from_str)){
                    eval(tmp_dyn_table_copy_from_str + "= '';");
                }
                eval(tmp_dyn_table_copy_to_str + "=" + tmp_dyn_table_copy_from_str + ";");

                tmp_dyn_table_copy_from_str = "z_table_" + j + ".doc_type[" + k + "]";
                tmp_dyn_table_copy_to_str = "z_table_readonly_" + j + ".doc_type[" + k + "]";
                if(!eval(tmp_dyn_table_copy_from_str)){
                    eval(tmp_dyn_table_copy_from_str + "= '';");
                }
                eval(tmp_dyn_table_copy_to_str + "=" + tmp_dyn_table_copy_from_str + ";");

                tmp_dyn_table_copy_from_str = "z_table_" + j + ".doc_version[" + k + "]";
                tmp_dyn_table_copy_to_str = "z_table_readonly_" + j + ".doc_version[" + k + "]";
                if(!eval(tmp_dyn_table_copy_from_str)){
                    eval(tmp_dyn_table_copy_from_str + "= '';");
                }
                eval(tmp_dyn_table_copy_to_str + "=" + tmp_dyn_table_copy_from_str + ";");
            }
        }
        enter("?");
}
```



```
//Function to clear all table content
function clearTableContent(){
    onscreen "*"
        var tmp_dyn_doc_str = "";

        for(var j=1; j<=z_table_counter; j++){
            tmp_dyn_doc_str = "z_table_" + j + ".doc_no";
            eval(tmp_dyn_doc_str + "= [];");        //Set each column of table becomes a blank array

            tmp_dyn_doc_str = "z_table_" + j + ".doc_part";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_" + j + ".doc_type";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_" + j + ".doc_version";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_readonly_" + j + ".doc_no";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_readonly_" + j + ".doc_part";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_readonly_" + j + ".doc_type";
            eval(tmp_dyn_doc_str + "= [];");

            tmp_dyn_doc_str = "z_table_readonly_" + j + ".doc_version";
            eval(tmp_dyn_doc_str + "= [];");
        }
        enter("?");
}
```