

Cores e Tamanhos PHC

Português

Mapeamentos

A utilização da opção de cores e tamanhos no MSS com dados do PHC, carece que sejam efetuadas as seguintes alterações aos mapeamentos das tabelas em baixo:

Tabela de Artigos (ART)

SQL 1

Mapeamentos

Campo	Descrição	Tabela ERP	Campo ERP
ARTTRD	Tipo de Recolha de dados	%OPTIONS%	Campo a Validar: texteis SE=1 ENTAO 'G' SENAO 'S'
ARTTGR	Tem Grelha	%OPTIONS%	Campo a Validar: texteis SE=1 ENTAO 'S' SENAO 'N'

SQL 2

Igual ao SQL 1, só alterando os seguintes dados:

Mapeamentos

Campo	Descrição	Tabela ERP	Campo ERP
ARTCOD	Código	%DEFAULT%	RTRIM(st.ref) + '.' + RTRIM(SX.cor) + '.' + RTRIM(SX.TAM)
ARTPUN	Preço Unitário	%OPTIONS%	Campo a Validar: sx.epv1 SE=0 ENTAO st.epv1 SENAO sx.epv1
ARTPU2	Preço Unitário 2	%OPTIONS%	Campo a Validar: sx.epv2 SE=0 ENTAO st.epv2 SENAO sx.epv2
ARTPU3	Preço Unitário 3	%OPTIONS%	Campo a Validar: sx.epv3 SE=0 ENTAO st.epv3 SENAO sx.epv3
ARTPU4	Preço Unitário 4	%OPTIONS%	Campo a Validar: sx.epv4

			SE=0 ENTAO st.epv4 SENAO sx.epv4
ARTTRD	Tipo de Recolha de dados	%DEFAULT%	'F'
ARTPM1	Preço Mínimo 1	%DEFAULT%	0
ARTPCU	Preço de Custo	%OPTIONS%	Campo a Validar: sx.epcusto SE=0 ENTAO st.epcusto SENAO sx.epcusto
ARTTGR	Tem Grelha	%OPTIONS%	Campo a Validar: Texteis SE=1 ENTAO 'S' SENAO 'N'

JOINS

Tipo	Tabela Detail	Condição
JOIN	sx	sx.ref=st.ref

Tabela de Atualização de Stock (ARTEX1)

SQL 1

Acrescentar este filtro:

Filtro
AND texteis = 0

SQL 2

Tabela Master: sx

Mapeamentos

Campo	Descrição	Tabela ERP	Campo ERP
ARTCOD	Código	%DEFAULT%	RTRIM(sx.ref) + '.' + RTRIM(sx.cor) + '.' + RTRIM(sx.tam)
ARTEX1	Stock	%DEFAULT%	sx.stock - sx.rescat + sx.qttrec

FILTRO

Filtro

sx.armazem = [VENDARMS] AND ([ARTEXT] = 1)

Tabela Grelha Horizontal (GRH)

SQL em modo Avançado

SQL

```
DECLARE @POSICAO INT
DECLARE @SQL VARCHAR(MAX)
DECLARE @REF VARCHAR(60)
DECLARE @REFT VARCHAR(60)
DECLARE @TAM VARCHAR(25)
DECLARE @MSGRH TABLE (
    REF VARCHAR(60) not null,
    DESCR VARCHAR(100) NOT NULL,
    DESCRV VARCHAR(100) NOT NULL,
    CC1 VARCHAR (30) NOT NULL,
    DC1 VARCHAR (100) NOT NULL,
    CC2 VARCHAR (30) NOT NULL,
    DC2 VARCHAR (100) NOT NULL,
    CC3 VARCHAR (30) NOT NULL,
    DC3 VARCHAR (100) NOT NULL,
    CC4 VARCHAR (30) NOT NULL,
    DC4 VARCHAR (100) NOT NULL,
    CC5 VARCHAR (30) NOT NULL,
    DC5 VARCHAR (100) NOT NULL,
    CC6 VARCHAR (30) NOT NULL,
    DC6 VARCHAR (100) NOT NULL,
    CC7 VARCHAR (30) NOT NULL,
    DC7 VARCHAR (100) NOT NULL,
    CC8 VARCHAR (30) NOT NULL,
    DC8 VARCHAR (100) NOT NULL,
    CC9 VARCHAR (30) NOT NULL,
    DC9 VARCHAR (100) NOT NULL,
    CC10 VARCHAR (30) NOT NULL,
    DC10 VARCHAR (100) NOT NULL,
    CC11 VARCHAR (30) NOT NULL,
    DC11 VARCHAR (100) NOT NULL,
    CC12 VARCHAR (30) NOT NULL,
    DC12 VARCHAR (100) NOT NULL,
    CC13 VARCHAR (30) NOT NULL,
    DC13 VARCHAR (100) NOT NULL,
    CC14 VARCHAR (30) NOT NULL,
    DC14 VARCHAR (100) NOT NULL,
    CC15 VARCHAR (30) NOT NULL,
```

```
DC15 VARCHAR (100) NOT NULL,
CC16 VARCHAR (30) NOT NULL,
DC16 VARCHAR (100) NOT NULL,
CC17 VARCHAR (30) NOT NULL,
DC17 VARCHAR (100) NOT NULL,
CC18 VARCHAR (30) NOT NULL,
DC18 VARCHAR (100) NOT NULL,
CC19 VARCHAR (30) NOT NULL,
DC19 VARCHAR (100) NOT NULL,
CC20 VARCHAR (30) NOT NULL,
DC20 VARCHAR (100) NOT NULL,
CC21 VARCHAR (30) NOT NULL,
DC21 VARCHAR (100) NOT NULL,
CC22 VARCHAR (30) NOT NULL,
DC22 VARCHAR (100) NOT NULL,
CC23 VARCHAR (30) NOT NULL,
DC23 VARCHAR (100) NOT NULL,
CC24 VARCHAR (30) NOT NULL,
DC24 VARCHAR (100) NOT NULL,
CC25 VARCHAR (30) NOT NULL,
DC25 VARCHAR (100) NOT NULL,
CC26 VARCHAR (30) NOT NULL,
DC26 VARCHAR (100) NOT NULL,
CC27 VARCHAR (30) NOT NULL,
DC27 VARCHAR (100) NOT NULL,
CC28 VARCHAR (30) NOT NULL,
DC28 VARCHAR (100) NOT NULL,
CC29 VARCHAR (30) NOT NULL,
DC29 VARCHAR (100) NOT NULL,
CC30 VARCHAR (30) NOT NULL,
DC30 VARCHAR (100) NOT NULL,
DAC VARCHAR (20) NOT NULL)
```

```
DECLARE ref_cursor CURSOR FOR SELECT RTRIM(SX.REF), RTRIM(SX.TAM) FROM SX (NOLOCK)
INNER JOIN sgt (NOLOCK) ON SGT.ref = SX.ref and sgt.tam = sx.tam GROUP BY SX.REF, SX.TAM,
SGT.pos order by sx.ref, SGT.pos
OPEN ref_cursor
FETCH NEXT FROM ref_cursor INTO @REF, @TAM
SET @SQL = "
SET @REFT = "
SET @POSICAO = 1
WHILE @@FETCH_STATUS = 0
BEGIN
    IF (@REFT <> @REF)
    BEGIN
        SET @REFT = @REF

        IF @SQL = "
            SET @SQL = 'SELECT ''' + @REFT + ''', "Tamanhos", "Cores"'
```

```

ELSE
BEGIN
    WHILE @POSICAO <= 30
    BEGIN
        SET @SQL = @SQL + ', ', ''
        SET @POSICAO = @POSICAO + 1
    END

SET @POSICAO = 1
SET @SQL = @SQL + ', '

--PRINT @SQL
INSERT INTO @MSGRH exec (@SQL)
SET @SQL = 'SELECT ' + @REFT + ', "Tamanhos", "Cores"'
END
END

IF (@TAM <> '') AND (@POSICAO <= 30)
BEGIN
    SET @SQL = @SQL + ', ' + @TAM + ', "TAM ' + @TAM + '"'
    SET @POSICAO = @POSICAO + 1
END
FETCH NEXT FROM ref_cursor INTO @REF, @TAM
END
CLOSE ref_cursor;
DEALLOCATE ref_cursor;

IF @REFT <> ''
BEGIN
    IF @POSICAO <= 30
    BEGIN
        WHILE @POSICAO <= 30
        BEGIN
            SET @SQL = @SQL + ', ', ''
            SET @POSICAO = @POSICAO + 1
        END

SET @POSICAO = 1
SET @SQL = @SQL + ', '

--PRINT @SQL
INSERT INTO @MSGRH EXEC (@SQL)
END
END

SELECT * FROM @MSGRH

```

Tabela Grelha Horizontal (GRV)

SQL em modo Avançado

SQL

```

DECLARE @POSICAO INT
DECLARE @INC INT
DECLARE @SQL VARCHAR(MAX)
DECLARE @REF VARCHAR(60)
DECLARE @REFT VARCHAR(60)
DECLARE @TAM VARCHAR(25)
DECLARE @TAMT VARCHAR(25)
DECLARE @COR VARCHAR(25)
DECLARE @CORT VARCHAR(25)
DECLARE @MSGRV TABLE (
    REF VARCHAR(60) not null,
    DESCR VARCHAR(100) NOT NULL,
    DESCRV VARCHAR(100) NOT NULL,
    SEQ INT NOT NULL,
    CC1 VARCHAR (1) NOT NULL,
    CC2 VARCHAR (1) NOT NULL,
    CC3 VARCHAR (1) NOT NULL,
    CC4 VARCHAR (1) NOT NULL,
    CC5 VARCHAR (1) NOT NULL,
    CC6 VARCHAR (1) NOT NULL,
    CC7 VARCHAR (1) NOT NULL,
    CC8 VARCHAR (1) NOT NULL,
    CC9 VARCHAR (1) NOT NULL,
    CC10 VARCHAR (1) NOT NULL,
    CC11 VARCHAR (1) NOT NULL,
    CC12 VARCHAR (1) NOT NULL,
    CC13 VARCHAR (1) NOT NULL,
    CC14 VARCHAR (1) NOT NULL,
    CC15 VARCHAR (1) NOT NULL,
    CC16 VARCHAR (1) NOT NULL,
    CC17 VARCHAR (1) NOT NULL,
    CC18 VARCHAR (1) NOT NULL,
    CC19 VARCHAR (1) NOT NULL,
    CC20 VARCHAR (1) NOT NULL,
    CC21 VARCHAR (1) NOT NULL,
    CC22 VARCHAR (1) NOT NULL,
    CC23 VARCHAR (1) NOT NULL,
    CC24 VARCHAR (1) NOT NULL,
    CC25 VARCHAR (1) NOT NULL,
    CC26 VARCHAR (1) NOT NULL,
    CC27 VARCHAR (1) NOT NULL,
    CC28 VARCHAR (1) NOT NULL,
    CC29 VARCHAR (1) NOT NULL,
    CC30 VARCHAR (1) NOT NULL,
    TES VARCHAR (1) NOT NULL)
    
```



```

DECLARE ref_cursor CURSOR FOR SELECT RTRIM(SX.REF), RTRIM(SX.cor), RTRIM(SX.TAM)
FROM SX (NOLOCK) INNER JOIN sgt (NOLOCK) ON sgt.ref = SX.ref AND SGT.tam = SX.tam
GROUP BY SX.REF, SX.COR, SX.TAM, SGT.pos ORDER BY sx.ref, sx.cor, SGT.pos
OPEN ref_cursor
FETCH NEXT FROM ref_cursor INTO @REF, @COR, @TAM
SET @SQL = ""
SET @REFT = ""
SET @CORT = ""
SET @POSICAO = 1
WHILE @@FETCH_STATUS = 0
BEGIN
    IF (@REFT <> @REF)
    BEGIN
        SET @REFT = @REF
        SET @CORT = @COR
        SET @INC = 1
        IF @SQL = ""
        BEGIN
            SET @SQL = 'SELECT ''' + @REFT + ''', ''' + @CORT + ''', "COR ' + @CORT + ''', '
            + RTRIM(CAST(@INC AS CHAR(3)))
            DECLARE ref_cursorTamanhos SCROLL CURSOR FOR SELECT RTRIM(SX.TAM)
            FROM SX (NOLOCK) INNER JOIN sgt (NOLOCK) ON SGT.ref = SX.ref and
            sgt.tam = sx.tam WHERE sx.ref = @REFT GROUP BY SX.REF, SX.TAM, SGT.pos
            order by sx.ref, SGT.pos
            OPEN ref_cursorTamanhos
        END
    ELSE
    BEGIN
        CLOSE ref_cursorTamanhos;
        DEALLOCATE ref_cursorTamanhos;
        WHILE @POSICAO <= 30
        BEGIN
            SET @SQL = @SQL + ', "N"'
            SET @POSICAO = @POSICAO + 1
        END
    END

    SET @POSICAO = 1
    SET @SQL = @SQL + ', ""'

    --PRINT @SQL
    INSERT INTO @MSGRV exec (@SQL)
    SET @SQL = 'SELECT ''' + @REFT + ''', ''' + @CORT + ''', "COR ' + @CORT + ''', ' +
    RTRIM(CAST(@INC AS CHAR(3)))
    DECLARE ref_cursorTamanhos SCROLL CURSOR FOR SELECT RTRIM(SX.TAM) FROM SX
    (NOLOCK) INNER JOIN sgt (NOLOCK) ON SGT.ref = SX.ref and sgt.tam = sx.tam WHERE
    sx.ref = @REFT GROUP BY SX.REF, SX.TAM, SGT.pos order by sx.ref, SGT.pos
    OPEN ref_cursorTamanhos
    END
    END

```



```

IF (@CORT <> @COR)
BEGIN
    SET @CORT = @COR

WHILE @POSICAO <= 30
BEGIN
    SET @SQL = @SQL + ', "N"'
    SET @POSICAO = @POSICAO + 1
END

SET @POSICAO = 1
SET @INC = @INC + 1
SET @SQL = @SQL + ', ""'

--PRINT @SQL
INSERT INTO @MSGRV exec (@SQL)
SET @SQL = 'SELECT "' + @REFT + '", "' + @CORT + '", "COR ' + @CORT + '", ' +
RTRIM(CAST(@INC AS CHAR(3)))

END

IF @TAM <> ""
BEGIN
    DECLARE @IPOST INT = 1
    FETCH FIRST FROM ref_cursorTamanhos INTO @TAMT
    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF ((@POSICAO <= @IPOST) AND (@POSICAO <= 30))
        BEGIN
            IF (@TAM = @TAMT)
            BEGIN
                SET @SQL = @SQL + ', "S"'
                FETCH LAST FROM ref_cursorTamanhos INTO @TAMT
            END
        END
        ELSE
        BEGIN
            SET @SQL = @SQL + ', "N"'
        END
        SET @POSICAO = @POSICAO + 1
        END
        ELSE IF (@POSICAO > 30)
        BEGIN
            FETCH LAST FROM ref_cursorTamanhos INTO @TAMT
            FETCH NEXT FROM ref_cursorTamanhos INTO @TAMT
            SET @IPOST = @IPOST + 1
        END
        END
        FETCH NEXT FROM ref_cursor INTO @REF, @COR, @TAM
    END
CLOSE ref_cursor;

```

```
DEALLOCATE ref_cursor;  
CLOSE ref_cursorTamanhos;  
DEALLOCATE ref_cursorTamanhos;  
  
IF @REFT <> ''  
BEGIN  
    WHILE @POSICAO <= 30  
    BEGIN  
        SET @SQL = @SQL + ', "N"'  
        SET @POSICAO = @POSICAO + 1  
    END  
    SET @SQL = @SQL + ', '''''  
  
    --PRINT @SQL  
    INSERT INTO @MSGRV EXEC (@SQL)  
END  
  
SELECT * FROM @MSGRV ORDER BY REF  
  
DELETE @MSGRV
```

Stored Procedures

Nos clientes já em funcionamento, deve ser acrescentado o SP “SPMSS_GetCorTamanho”, e alterados os SP “SPMSS_DocDocFactLin” e “SPMSS_DocDossInternLin”.

- SPMSS_GetCorTamanho

```
CREATE PROCEDURE [dbo].[SPMSS_GetCorTamanho]
@Ref VARCHAR(60),
@r varchar(60) OUTPUT,
@c varchar(25) OUTPUT,
@t varchar(25) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    set @r = SUBSTRING(@ref, 1, CHARINDEX('.', @ref) - 1)
    set @c = SUBSTRING(@ref, CHARINDEX('.', @ref) + 1, DATALENGTH(@ref) -
DATALENGTH(@r))
    set @c = SUBSTRING(@c, 1, CHARINDEX('.', @c) - 1)
    set @t = SUBSTRING(@ref, DATALENGTH(@r) + DATALENGTH(@c) + 3,
DATALENGTH(@ref))
END
GO
```

- SPMSS_DocDocFactLin

Acrescentar ao SP já existente antes de EXECUTE SPMSS_GetDadosDoc

```
-- Verifica se é um artigo com cor e tamanho
SET @refCorTam = dbo.ExtractFromACL(@DCLACL, 15)
IF (@refCorTam = 'F')
BEGIN
    SET @refCorTam = @ref
    EXECUTE SPMSS_GetCorTamanho @refCorTam, @ref OUTPUT, @Cor OUTPUT, @Tam
    OUTPUT
    SET @Texteis = 1
END
ELSE
BEGIN
    SET @COR = ''
    SET @Tam = ''
    SET @Texteis = 0
END
```

- SPMSS_DocDocFactLin

Acrescentar ao SP já existente antes de EXECUTE SPMSS_GetDadosDoc

```
-- Verifica se é um artigo com cor e tamanho
SET @refCorTam = dbo.ExtractFromACL(@DCLACL, 15)
IF (@refCorTam = 'F')
BEGIN
    SET @refCorTam = @ref
    EXECUTE SPMSS_GetCorTamanho @refCorTam, @ref OUTPUT, @Cor OUTPUT, @Tam
    OUTPUT
    SET @Texteis = 1
END
ELSE
BEGIN
    SET @COR = ''
    SET @Tam = ''
    SET @Texteis = 0
END
```