**Actiuni asociate machetelor**

Pentru a fi mai usor de inteles cum functioneaza facilitatile de design in zona de actiuni asociate cu machetele, vom porni de la exemple practice, enumerand enunturi de probleme si exemplificand solutia de rezolvare.

Va solicit toata ingaduita pentru exemplele care sunt uneori cam puerile,  dar important e sa intelegeti functionarea.
Pe baza acestor exemple veti putea dezvolta singuri proceduri proprii, pe caz real, dupa nevoi.

Pe factura de iesire vom asocia, ca cerinte, urmatoarele actiuni ( acceptati enunturile banale fara sa judecati!):

1. **validare la salvarea facturilor, astfel incat fiecare articol sa aiba clasa de articol asociata;**
2. **atentionare daca pe factura sunt linii cu Cantitate < 3;**
3. **posibilitatea de a afisa o lista cu valorile facturate  pe gestiuni;**
4. **posibilitatea de a afisa o lista cu codurile externe si cele interne pentru fiecare articol din factura.**
5. **import rezervare pe factura de iesire.**

Pentru ultima problema sunt necesare niste explicatii suplimentare.

Presupunem  ca firma lucreaza cu rezervari in domeniul turismului.

Rezervarile contin informatii legate de servicii care urmeaz a fi prestate si despre pachetele de turism oferite.

Tabelele in care se vor salva datele sunt deschise la optiunea "Machete proprii" si nu le voi explica aici.
Folosind optiunea de design, macheta pentru introducerea datelor la rezervari este deja configurata si arata ca in imaginile de mai jos.





Pornim sa aplicam aceste cerinte formulate mai sus pe factura de iesire.
Iata cum arata macheta de factura, cu toate elementele care rezolva cerintele afisate la design:


Elementele incercuite sunt controalele noi adaugate pe macheta originala.
Pentru a le face loc, am ascuns alte controale care presupunem ca sunt neutilizate de firma.
Pentru a vizualiza ce controale am ascuns, apas butonul "Show/Hide invisible", cel incercuit pe Object inspector in imaginea de mai jos:


Efectul este urmatorul:


Observati controalele marcate cu patrat galben la care s-a setat proprietatea Visible = False.

Rezolvarea punctelor 1..5 se face din design (F2 + buton "Design" pe o inregistrare noua de factura), pagina "Actiuni asociate".



Tipul de actiune pentru cele 5 enunturi este:
**1- validare**         (pentru validare la salvarea facturilor, astfel incat fiecare articol sa aiba clasa de articol asociata**);
2- atentionare**     (pentru atentionare daca pe factura sunt linii cu Cantitate < 3); **3-interogare** (pentru posibilitatea de a afisa o lista cu valorile facturate  pe gestiuni); **4- interogare** (pentru posibilitatea de a afisa o lista cu codurile externe si cele interne pentru fiecare articol din factura);
**5- import date**            (importat rezervare pe factura de iesire);
Pentru fiecare caz in parte se inregistreaza un SQL.  E important sa integeleti ce inseamna:
**a**. "SQL pe baza de date"   : acesta foloseste sintaxa  **Oracle/PosgreSQL** pentru a aduce date tabelele originale ( exclus tabelele temporare) din Ora sau PG.
**b**. "SQLite pe memorie"     : acesta foloseste sintaxa  **SQLite** aplicat pe tabelele de pe memorie, cele ale caror nume si structura le vedeti in optiunea "Tabele WME" pe document.

Mecanismul de lucru cu tabele in WMEnterprise, pentru a intelege diferenta intre cele doua tipuri de SQL, este urmatoarea pentru fiecare document:
- **Oracle/PosgreSQL** se executa fraze SQL care muta date din tabelele originale (ex: IESIRI) in tabelele temporare (ex: TMP\_IESIRI);
- **WME** aduce inregistrarile din tabelele temporare in niste tabele pe memorie, cu numele de tabela si de campuri asa cum le vedeti in "Tabele WME";
  tabelele pe memorie au structura identica cu tabelele temporare, dar au alte denumiri , si anume fara prefix TMP\_;
- tabelele de pe memorie astfel incarcate, stau in spatele tuturor controalelor vizuale de pe machete si pot fi editate dupa ce apasati "Adaug" sau "Modific";
- controalele prin care date sunt introduse la interfata memoreaza valorile in tabelele de  pe memorie;
- la optiunea "Salvez" se ruleaza algoritmi de atentionare / validare privitor la datele introduse;
- in cazul in care toate verificarile de corectitudine a datelor au trecut cu succes, **WME** trimite informatiile din tabelele de pe memorie inapoi in tabelele temporare din **Oracle sau PosgreSQL;**- dupa ce au ajuns in temporare (ex: TMP\_IESIRI), acestea se codifica unic si ajung in cele din urma sa fie integrate in tabelele originale (ex: IESIRI).

Se intelege ca datele prezente pe memorie intainte de salvare (inainte de a ajunge in **Ora/PG**) nu au cum sa fie interogate cu fraze SQL clasice care se aplica doar pentru tabele **Oracle sau PosgreSQL.**Pentru datele pe memorie se aplica fraze SQL specifice, cu sintaxa de la un motor de baza de date special conceput pentru lucru pe memorie, denumit **SQLite**. **E de inteles ca in SQLite nu au ce cauta tabele din Ora sau PG. acestea nefiind incarcate pe memorie!
SQL pe baza de date folosete excusiv tabele din baza de date Ora/PG,  iar SQLite pe memorie foloseste exclusiv tabelele din modulul de date (vizualizate prin "Tabele WME").
Cu alte cuvinte vorbim despre 2 dimensiuni paralele, care nu au nici o legatura una cu alta!

Acum, pe rand, voi explica fiecare rezolvare.

1- Validare la salvarea facturilor, astfel incat fiecare articol sa aiba clasa de articol asociata.** **Mesaj validare:** <DenObiect> nu are clasa declarata!*<DenObiect> reprezinta campul pentru denumirea articolului care nu trece validarea, asa cum e el pe memorie in inregistarea din tabela IESIRI1.*

**SQL pe baza de date**

**Select O.Denumire DenObiect
From NartObj o
inner  join NArt a on a.codarticol = o.codarticol
Where Coalesce(a.Clasa, 0) = 0 and O.CodObiect = <Iesiri1.CodObiect>***Acest SQL este lansat in Ora/PG, dupa ce  <Iesiri1.CodObiect> este inlocuit cu CodObiect din linia tabelei de pe memorie Iesiri1.**Daca in SQL pe baza de date este folosita tabela de detaliu principala de pe memorie, in cazul de fata Iesiri1, atunci acest SQL se va executa pentru fiecare linie de pe factura, deci pentru fiecare linie din Iesiri1.
Similar daca in SQL este folosita tabela de EXTENSIE la detaliul principal de pe memorie, in cazul de fata Iesiri1EXT.*
**2- Atentionare daca pe factura sunt linii cu Cantitate < 3** **Mesaj atentionare:**<DenObiect>: Cantitate facturata < 3 ( <Cant> )
*<DenObiect> reprezinta campul pentru denumirea articolului care nu trece validarea, asa cum e el pe memorie in inregistarea din tabela IESIRI1.*
 *<Cant> reprezinta campul pentru cantitate la articolul care nu trece validarea, asa cum e el pe memorie in inregistarea din tabela IESIRI1.*

**SQLite pe menorie**

**Select I1.DenObiect, I1.Cant
From Iesiri1 I1
Where I1.Cant < 3***Acest SQL e lansat in SQLite pe tabelele pe memorie care nu sunt inca in baza de date.*
**3- Afisare lista cu valorile facturate  pe gestiuni** **Titlu macheta de afisare:**Total valori pe gestiuni*Textul e afisat ca titlu pe macheta care afiseaza grila cu valorile*
 **SQLite pe memorie**

 **Select Rez.Gestiune, Sum(Rez.V) Valoare**

**from**

**(
Select Coalesce(Ix1.DenGest, Lt.Gestiune) Gestiune, I1.Cant\*I1.PU V
from
Iesiri1 I1
left join Iesiri1Ext Ix1 on Ix1.CodIes1 = I1.CodIes1
left join
( select
  Coalesce(x.CodIesInt1, 0) CodIes1,
  Case when x.Gest = 0
    then Null
    else Substr(x.DenGestIoArt, 1, InStr(x.DenGestIoArt, Chr(13))-1)
  end Gestiune
  from LivrTot x
) Lt on Lt.CodIes1 = I1.CodIes1
) Rez
group by Rez.Gestiune**

*Acest SQL e lansat in SQLite pe tabelele pe memorie care nu sunt inca in baza de date.
De observat cum am extras gestiunea, deoacere aveam urmatoarele probleme.

Pe Iesiri1 nu exista camp cu denumirea de gestiune asociata serviciilor!
Mi-am "procurat" singur acest camp pe extensia la Iesiri1!
Cu alte cuvinte, am adaugat un camp temporar pe extensia Iesiri1EXT : DenGest.
Iesiri1EXT.DenGest se actualizeaza la schimbarea valorii Iesiri1.GestAct (vezi structura Iesiri1EXT!).

A doua problema este faptul ca pe LivrTot (pagina de livrare din factura) exista un camp ce contine Denumirea gestiunii + Denumirea tipului contabil.
Am observat ca cele 2 denumiri sunt separate prin caracterul  CR, adica #13.
Asadar, pentru liniile de stoc, care au gestiunea pe campul LivrTot.DenGestIOArt, am copiat textul pana la #13 (adica Chr(13) in SQL).*
**4- Afisare lista cu codurile externe si cele interne pentru fiecare articol din factura.** **Titlu macheta de afisare:**Coduri articole*Textul e afisat ca titlu pe macheta care afiseaza grila cu valorile*
 **SQL pe baza de date**

**Select
distinct O.Denumire, O.CodExtern, O.CodIntern
From NartObj o
Where O.CodObiect = <Iesiri1.CodObiect>***Atentie! SQL pe baza de date are intotdeauna o legatura cu un camp de pe memorie, acesta facand legatura cu datele introduse pe macheta care inca nu se gasesc in Ora/PG.*

*Acest SQL este lansat in Ora/PG, dupa ce  <Iesiri1.CodObiect> este inlocuit cu CodObiect din linia tabelei de pe memorie Iesiri1.**Daca in SQL pe baza de date este folosita tabela de detaliu principala de pe memorie, in cazul de fata Iesiri1, atunci acest SQL se va executa pentru fiecare linie de pe factura, deci pentru fiecare linie din Iesiri1.
Similar daca in SQL este folosita tabela de EXTENSIE la detaliul principal de pe memorie, in cazul de fata Iesiri1EXT.***5- Import rezervare pe factura de iesire.**   **Titlu macheta de import:**
neutilizat
 **SQL pe baza de date**

 **select R1.CODOBIECT, R1.CANTITATE as Cant, R1.PRET as PU,
       R1.CodRezervare1, R1.Observatii as ObsRezervare,
       R.DataDoc as DataRezervare, R.CodPart as Partener
from rezervare R
inner join Rezervare1 R1 on R1.CodRezervare = R.CodRezervare
where R.CodRezervare = <IesiriExt.CodRezervare>**

*Atentie! SQL pe baza de date are intotdeauna o legatura cu un camp de pe memorie, acesta facand legatura cu datele introduse pe macheta care inca nu se gasesc in Ora/PG.*

*Acest SQL este lansat in Ora/PG, dupa ce  <IesiriExt.CodRezervare> este inlocuit cu CodRezervare din linia tabelei extensie la Iesiri de pe memorie, IesiriEXT.**Desigur, pentru a avea acest camp in IesiriEXT, l-am adaugat in structura extensiei la Iesiri.**El poate fi camp original (se va si salva pe tabela din Ora/PG IesiriEXT!), sau doar temporar, caz in care este memorat doar pe TMP\_IesiriEXT(caz in care nu se salveaza pentru aceasta inregistrare in IesiriEXT);
Acest camp va fi completat folosind un control de alegere, denumit buton Lookup, asa cum o sa vedeti mai jos, iar imediat dupa alegerea rezervarii  se va executa SQL de import, acesta aducand liniile rezervarii pe Iesiri1.*

*Observatie
Cursorul obtinut la executia SQL are urmatoarele campuri:* ***CODOBIECT, CANT, PU,  CodRezervare1, ObsRezervare, DataRezervare, Partener****.
Algoritmul e suficient de sofisticat si foloseste campurile astfel:
\*****CODOBIECT, CANT, PU*** *le gaseste in structura IESIRI1, drept care adauga o linie noua in Iesiri1 (implicit si in Iesiri1EXT care e sincron cu Iesiri1);
\*****CodRezervare1, ObsRezervare*** *le gaseste pe Iesiri1EXT (extensia detaliului principal), drept care le pune acolo;
  CodRezervare1 e camp original, se va salva in Ora/PG pentru a avea evidenta liniilor importate;
  ObsRezervare e camp temporar, se incarca doar la interfata pentru a da informatii legate de linia importata din rezervari;
\*****DataRezervare*** *il gaseste in extensia IesiriEXT, fiind un camp temporar adaugat de mine;
  Se incarca doar la interfata pentru a da informatii legate de rezervare;
\*Algoritmul identifica un buton Lookup pe macheta de iesiri care e legat la campul Partener (butonul "Client");*  ***Partener*** *e campul standard in Iesiri si odata identificat, algoritmul il propune pentru alegere pe view mic de parteneri,  actionand  automat butonul "Client"*De remarcat in grila coloana  ActionID introdusa la import (91) si la interogari (50, 51).
ActionID se vor lega de butoanele care vor executa importul sau interogarea.

De asemenea, de remarcat pe pagina de actiuni asociate un butonul "Test SQL", care desigur va permite sa testati fraza SQL pe care ati produs-o!

Actiunea de import necesita alegerea in prealabil a unui cod, in cazul nostru al unui cod de rezervare.
Fiind vorba de o alegere, vom folosi pentru aceasta controlul potrivit, si anume un LookupButton.Observati proprietatea ActionID = 90.
Aceasta proprietate va declansa SQL-ul atasat liniei de import cu ActionID 90 si va aduce datele pe factura.
Butonul e legat la Datasource = DSExtMaster (adica in final la IesiriEXT).
Codul ales se stocheaza in DataField =  CodRezervare , adica IesiriEXT.CodRezervare.

Pentru a afisa pe buton informatii  legate de rezervare (altfel CodRezervare nu ar avea inteles pentru operator), se foloseste LookupSource = DcCust\_1012.
Acest mod de codificare a unui view mic asociat, DcCust\_1012, identifica prin 1012 cod de identificare a machetei proprii "rezervari" , care are ID egal cu 1012.
Pentru a aduce informatiile cerute in LookupDisplay, si anume SERIADOC;NrDOC;DataDoc (care sunt din tabela Rezervari!), butonul leaga campul IesiriEXT.CodRezervare
cu inregistrarea din DcCust\_1012 (care e tabela Rezervari) prin LookupField = CodRezervare ( adica Rezervari.CodRezervare).
**Rezervari.CodRezervare = IesiriEXT.CodRezervare,** unde Rezervari.CodRezervare este **LookupField**, iar IesiriEXT.CodRezervare este **DataField**.

De pe aceasta lini afiseaza pe butonul lookup informatiile urmatoare:


ActionID 50 din grila de "Actiuni asociate" e legat de un buton simplu, care are si el proprietate ActionID potrivita.


Similar, ActionID 51 din grila de "Actiuni asociate" e legat de un buton simplu, care are si el proprietate ActionID potrivita.


O ultima observatie legata de butoanele simple.


Exista la butoanele simple o proprietate denumita DataSource.

\*Daca aici NU e specificata o valoare,  atunci butonul e activ indiferent daca macheta e in editare sau inregistrarea a fost salvata.
\*Daca aici e specificata o valoare,  atunci butonul e activ astfel:
- cand InvertDSState = No, atunci butonul e activ doar cand esti in editate pe macheta;
- cand InvertDSState = Yes, atunci butonul e activ doar dupa salvarea inregistrarii;