

Implementarea tehnologiei client/server utilizând SPT într-o aplicație de evidența clienți

Lect.dr. Cornelia MUNTEAN

Facultatea de Științe Economice, Universitatea de Vest Timișoara

This paper outlines the advantages of client/server technology and especially of MySQL, a relational database management system (RDBMS). It is exemplified the configuration of the datasource and of the ODBC driver for the MySQL database engine for a simple application of invoiced sums for specific clients. There are also presented the main instructions specific for SQL PassThrough (SPT) technology in Visual FoxPro, the environment in which the client application was implemented.

Keywords: *client/server technology, MySql.*

Despre avantajele tehnologiei client/server și a SGBDR-ului MySQL

În majoritatea sistemelor client/server, rolul calculatorului client este de a asigura o interfață grafică cu utilizatorul (GUI), astfel încât utilizatorul să poată avea acces cu ușurință la sistem. Interfața GUI este de obicei Windows, dar poate fi și OS/2, Unix Motif sau chiar Macintosh. Totuși, interfața GUI nu reprezintă o cerință a modelului client/server.

Într-o aplicație client/server tipică, calculatorul client asigură interfața GUI, iar calculatorul server asigură accesul la baza de date prin intermediul unui server SQL. Fără tehnologia client/server pentru o aplicație de baze de date, calculatorul trebuie să analizeze fiecare înregistrare doar pentru a selecta înregistrările care corespund unui criteriu de căutare și fiecare înregistrare trebuie transmisă prin rețea. În cazul client/server, calculatorul client trimite cererea SQL calculatorului server, care o prelucrează și înapoiază doar acele înregistrări care se potrivesc condițiilor solicitate.

În majoritatea sistemelor client/server, clientul se ocupă cu detaliile interfeței cu utilizatorul, iar serverul cu detaliile accesului la baza de date. Regulile afacerii însă, adică programul logic care formulează politica afacerii, pot fi implementate atât în porțiunea client, cât și în cea de server. Dacă sunt localizate pe server, regulile afacerii pot fi implementate prin intermediul programelor care apelează un mecanism de apel al procedurilor la distanță (RPC – Remote Procedure Call) sau proceduri stocate într-un sistem DBMS.

Programele care rulează pe calculatoarele client sunt, de obicei, realizate folosind unul dintre descendentii instrumentelor de programare vizuală (Borland Delphi, Microsoft Visual Basic, Visual Foxpro, C++, Powersoft PowerBuilder). Pentru aplicația client prezentată am ales Visual Foxpro și motorul de baze de date MySQL.

Motorul de baze de date MySQL a fost instalat pe un sistem de operare Windows XP, ca un serviciu prin comanda `mysqld-max-nt --install`. Acest serviciu se poate dezinstala folosind comanda `mysqld-max-nt --remove`. Sufixul `nt` arată că este vorba de SO Windows NT4, 2000 și XP și `max` arată că este un server care permite a se folosi baze de date care suportă lucrul cu tranzacții de tipul *innodb*. Toată configurarea serverului se specifică în fișierul `my.cnf`, care se amplasează în rădăcina.

Configurarea sursei de date și a driverului ODBC pentru MySQL

Protocolul folosit pentru implementarea modelului client/server este ODBC (Open DataBase Connectivity), protocol standard de comunicare cu serverele de baze de date. Driverul ODBC reprezintă programul care rulează în sistemul de calcul local și permite aplicației client să aibă acces la datele SQL utilizând un subset obișnuit al limbajului SQL, fără să trebuiască să știe care producător SQL DBMS îl va folosi (figura 1).

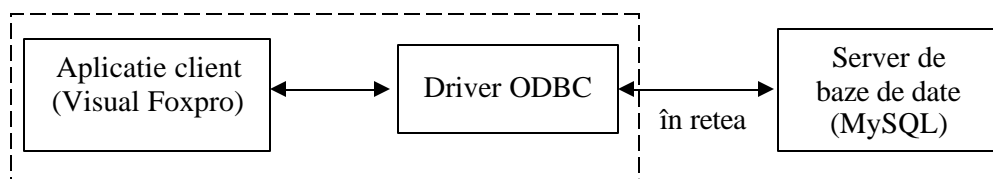


Fig. 1. Arhitectura client/server a unei aplicatii de baze de date

Din Visual Foxpro, driverul ODBC se vede ca o sursa de date de un anumit tip (în functie de tipul bazei de date). Operatia de configurare a surselor de date si a driverelor ODBC este externa mediului Visual Foxpro si tine de sistemul de operare (Windows). Daca s-a instalat interfața ODBC după instalarea SGBD-ului, în aplicatia Control Panel din Windows la Administrative Tools apare iconul Data Sources (ODBC) corespunzătoare aplicatiei de configurare a interfeței ODBC. Un dublu-click pe acest icon determina deschiderea ferestrei de dialog pentru configurarea driverelor ODBC si a surselor de date disponibile în aplicatii.

Fereastra contine mai multe pagini alternative. În pagina **Drivers** sunt prezentate driver-urile ODBC instalate în sistem. Dacă dorim sa folosim un anumit tip de baza de date (în cazul nostru MySQL), driverul corespunzător trebuie sa se afle în aceasta lista. În caz contrar trebuie obținut de la firma producătoare si trebuie instalat.

Pagina System DSN este folosita pentru configurarea surselor de date (figura 2). Fiecare sursa de date are la baza un driver ODBC. Butonul Add foloseste pentru adaugarea unei noi surse de date, iar butonul Remove pentru stergerea unei surse de date. În cazul de fata avem o singura sursa de date cu numele Fact_pred (facturi predate) de tipul MySQL. Configurarea sursei de date se face prin selectarea sursei din lista, urmata de actionarea butonului Configure. Fereastra de dialog care apare depinde de tipul sursei de date. De exemplu, fereastra din figura 3 corespunde lui MySQL, de care vom avea nevoie în aplicatia noastra.

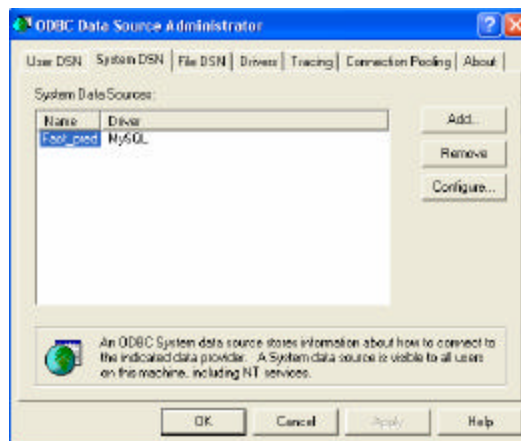


Fig. 2. Pagina System DSN din ODBC Data Source Administrator

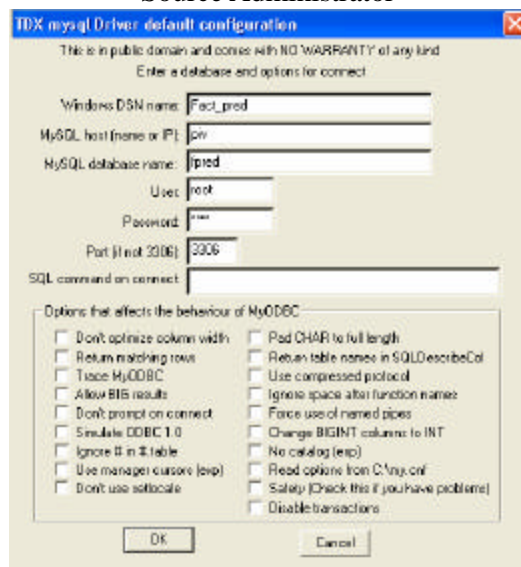


Fig. 3. Fereastra de configurare a sursei de date Fact_pred

În acest caz este necesara indicarea unor parametri specifici rețelei, precum sursei de date Windows (în acest caz *Fact_pred*), numele calculatorului gazda, unde este instalat MySQL (în acest caz *piv*), numele bazei de date MySQL la care se dorește conectarea (în acest caz *fpred*), numele utilizatorului si parola acestuia, portul si eventual comanda SQL care sa se execute la conectare.

Prezentarea aplicatiei

Se lucreaza cu doua tabele, care au fost create pe masina „server” (cea care stocheaza baza de date MySQL), folosind instructiuni SQL specifice (CREATE):

- tabela *fpred* (facturi predate), care contine o evidenta a facturilor clientilor, având câmpurile: NRC (numar curent), reprezinta contorul de înregistrari al bazei de date si este si cheie primara; GRUP – un câmp care reprezinta numarul grupului (între 1 si 6) în care se încadreaza contul dat în câmpul CONT; CONT – contine numarul contului; VAL – contine valoarea înregistrata în contul respectiv; DATA – reprezinta data facturarii.

```
create table fpred
(
  NRC          int unsigned not null auto_increment,
  GRUP        tinyint unsigned,
  CONT        smallint unsigned,
  VAL         int,
  DATA       Date default "0000-00-00",
  primary key (NRC),
  index (cont)
)
```

- tabela *grup*, este o tabela cu doar 6 înregistrari, date mai jos:

```
1, "CONTURI IN INTERVALUL", 1, 500
2, "CONTURI IN INTERVALUL", 501, 1000
3, "CONTURI IN INTERVALUL", 1001, 1500
4, "CONTURI IN INTERVALUL", 1501, 2000
5, "CONTURI IN INTERVALUL", 2001, 2500
6, "CONTURI IN INTERVALUL", 2501, 3000
```

si creata cu comanda SQL urmatoare:

```
create table grup
(
  GRUP          tinyint unsigned primary
  key ,
  DESCRIERE     Char(30),
  CONT1        Smallint unsigned,
  CONT2        Smallint unsigned
)
```

Prin aplicatia de mai jos, care am dorit sa fie cât mai usor de înteles, nu mi-am propus altceva decât sa exemplific modul de lucru al unei aplicatii cu baze de date client/server utilizând SQL PassThrough din Visual Foxpro, deci modalitatea SQL „înglobat” de a amesteca instructiunile SQL cu cele ale altor limbaje de programare. Aplicatia permite introducerea, cautarea, modificarea, stergerea si/sau listarea sumelor facturate pentru anumiti clienti. Acestia sunt identificati dupa un numar de cont, iar la listare aceste conturi sunt grupate în sase grupe descrise în tabela *grup*.

La pornire, aplicatia întreaba daca se doreste sau nu initializarea borderoului, permitând, la raspuns afirmativ, stergerea completa a continutului tablei fpred printr-o secventa SQL PassThrough ce va fi afisata mai jos, dupa care se va afisa fereastra de dialog din figura 4:

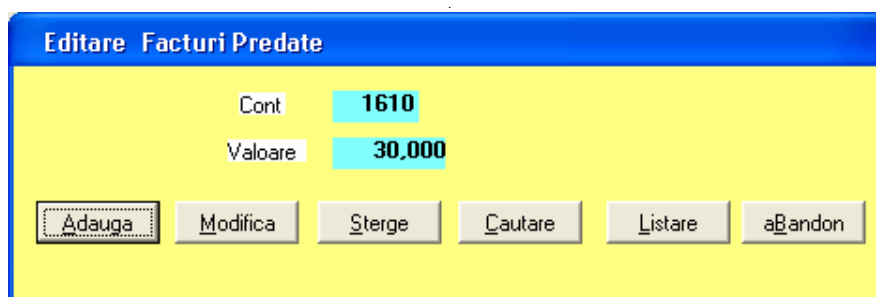


Fig. 4. Fereastra de dialog/meniu a aplicatiei Formal

Prin intermediul acesteia utilizatorul are posibilitatea introducerii, cautarii, modificarii stergerii si/sau listarii unor facturi pentru anumiti clienti. La selectarea butonului Listare apare fereastra de dialog din figura 5, care

permite listarea la imprimanta sau pe ecran a situatiei facturilor predate pentru o anumita data calendaristica si între anumite pagini limita.

Fig. 5. Fereastra de dialog în vederea listarii facturilor predate Forma2

Proiectul Visual Foxpro va contine un program principal, cele doua forme de mai sus cu codul aferent si un raport realizat cu gere-

ratorul de rapoarte, care va lista pe pagini diferite, cu subtota luri, conturile grupate pe cele sase grupe (figura 6).

S.N."ALFA" S.A.		BORDEROU		
Sucursala TIMISOARA		de facturi predate la CEC		
		pe luna:	anul:	2003
Page	1	Cont	Valoare	
Nr.crt				
1		101	12000	
2		107	1.212	
3		379	650	
		Subtotal :	13862	
...				
Page 6				
...				
		Total general:	330.262	

Fig. 6. Raportul pe grupe de conturi

Prin liniile comentariu inserate în cod poate fi înțeles mecanismul de functionare al aplicatiei, respectiv instructiunile SQL

PassThrough de dialog cu serverul de baze de date. Programul principal contine, printre alte instructiuni, urmatoarele linii de cod:

```
Proc Pfpred                && Facturi predate
#INCLUDE aplvfp.h          && pt. Apelul unor constante din Visual Foxpro
...
=SQLDISCONNECT(0)        && invalideaza toate conexiunile
*---conectare la sursa de date ODBC Fact_pred, de catre utilizatorul root cu parola
root
hcon1= SQLCONNECT( 'Fact_Pred', 'root','root')
* se verifica handler-ul de conexiune returnat de functia de conectare SQLCONNECT
if hcon1>0
  wait wind 'OK la conectare'+str( hcon1,4) nowait
  else
  wait wind 'EROARE la conectare'+str( hcon1,4)
  retu
endif
vprop= SQLSETPROP(hcon1, 'asynchronous', .F.)
Wait 'Doriti initializarea BORDEROULUI D/N=enter ?' to d2 wind
if upper(d2)='D'
* in variabila pcod se va memora de fiecare data fraza SQL de interogare trimisa
serverului
pcod= "delete from fpred" && sterge complet tabela fpred
vprep= SQLPREPARE(hcon1, pcod, 'cr1') && cr1 este numele cursorului
* cursor se numeste fisierul temporar unde sunt returnate rezultatele instructiunii
SELECT
```

```

vexec= SQLEEXEC ( hcon1, pcod)
=Fer(vexec) && anunta posibila eroare
endif
...
Function Fer          && in caz de eroare
param pexec
local laErrors[10] && local array (tablou local)
if pexec<0
    =AERROR (laErrors)
    lcErrorString = laErrors[1,2] + CHR(13) + CHR(10) + pcod
    =MESSAGEBOX (lcErrorString, MB_OK + MB_ICONSTOP, FATAL_ERROR_LOC)
* parametrii din MESSAGEBOX sunt constante predefinite din aplvfp.h
endif
return

```

Câteva instructiuni de cod aferente formei Formal sunt:

```

*vp este numele variabilei care comanda grupul de butoane de comanda din Formal
do case
case vp=1      && Adaugare -----
    xgrup= Fgrup(xcont) && cu litera de inceput x s-au notat variabile locale, cu v
    variabile globale
    pcod= "INSERT INTO fpred (cont, val, grup ) VALUES (?@xcont, ?@xval, ?@xgrup) "
* cu ?@variabila se noteaza parametrii de intrare in tehnologia SPT in Visual
Foxpro
    vprep= SQLPREPARE(hcon1, pcod)
    vexec= SQLEEXEC( hcon1)
    =Fer(vexec) && anunta posibila eroare
xcont=0
xval=0
thisform.refresh
_curobj=objnum(xcont)      && se focalizeaza xcont
case vp=2      && Modificare -----
    xgrup= Fgrup(xcont)
    pcod= "UPDATE fpred SET cont= ?@xcont, val=?@xval, grup= ?@xgrup ;
    WHERE nrc= ?@vnrc "
    vprep= SQLPREPARE(hcon1, pcod)
    vexec= SQLEEXEC ( hcon1)
    =Fer(vexec) && anunta posibila eroare
retu 0
case vp=3      && Stergere -----
    pcod= "DELETE FROM fpred WHERE nrc= ?@vnrc "
    vprep= SQLPREPARE(hcon1, pcod)
    vexec= SQLEEXEC ( hcon1)
    =Fer(vexec) && anunta posibila eroare
retu 0
case vp=4      && Cautare/ Rasfoire -----
    pcod= "SELECT nrc, cont, val FROM fpred order by cont "
    vprep= SQLPREPARE(hcon1, pcod, 'cursor')
    vexec= SQLEEXEC ( hcon1)
    =Fer(vexec) && anunta posibila eroare
select cursor
brow normal noedit fields cont, val title 'Pozitionati-va pe inreg. de modificat'
    vnrc= int(nrc)
wait 'vnrc:' +str(vnrc,3) wind
    pcod= "SELECT cont, val FROM fpred WHERE nrc= ?@vnrc "
    vprep= SQLPREPARE(hcon1, pcod, 'fpred')
    vexec= SQLEEXEC ( hcon1)
    =Fer(vexec) && anunta posibila eroare
select cursor&& preia datele de modificat
xcont = cont
xval =val
thisform.refresh
_curobj=objnum(xcont)
case vp=5      && Listare -----
do form Forma2
retu 0
case vp=6      && Abandon -----

```

```
thisform.release  
endcase
```

Pentru listare, liniile de cod aferente formei Forma2 sunt:

```
case vp=1                && Listare  
vexec= SQLEXEC( hcon1, " UPDATE fpred SET data= ?@vdata" )  
=Fer(vexec) && anunta posibila eroare  
pcod= "SELECT cont, val, grup from fpred order by cont"  
vprep= SQLPREPARE(hcon1, pcod, 'fpred')  
vexec= SQLEXEC( hcon1, pcod)  
=Fer(vexec) && anunta posibila eroare  
select fpred  
wait 'Imprimanta este pregatita D/N=enter ? ' wind to dl  
vfis= 'C:\TEMP\FPRED.TXT'  
if upper(dl)='D'  
    repo form rlisfpr to print noconsole range vb, ve  
else  
    repo preview form rlisfpr noconsole range vb, ve  
endif  
...  
endcase
```

Bibliografie

1. Paul DuBois, MySQL, Editura Teora, 2001
2. Doug Lowe, Tehnologia client/server pentru toti, Editura Teora, 1996

3. Gabriel Dima, Mihai Dima, Bazele Visual FoxPro 5.0, Editura Teora, 2001