

## Bazy danych w DELPHI (3)

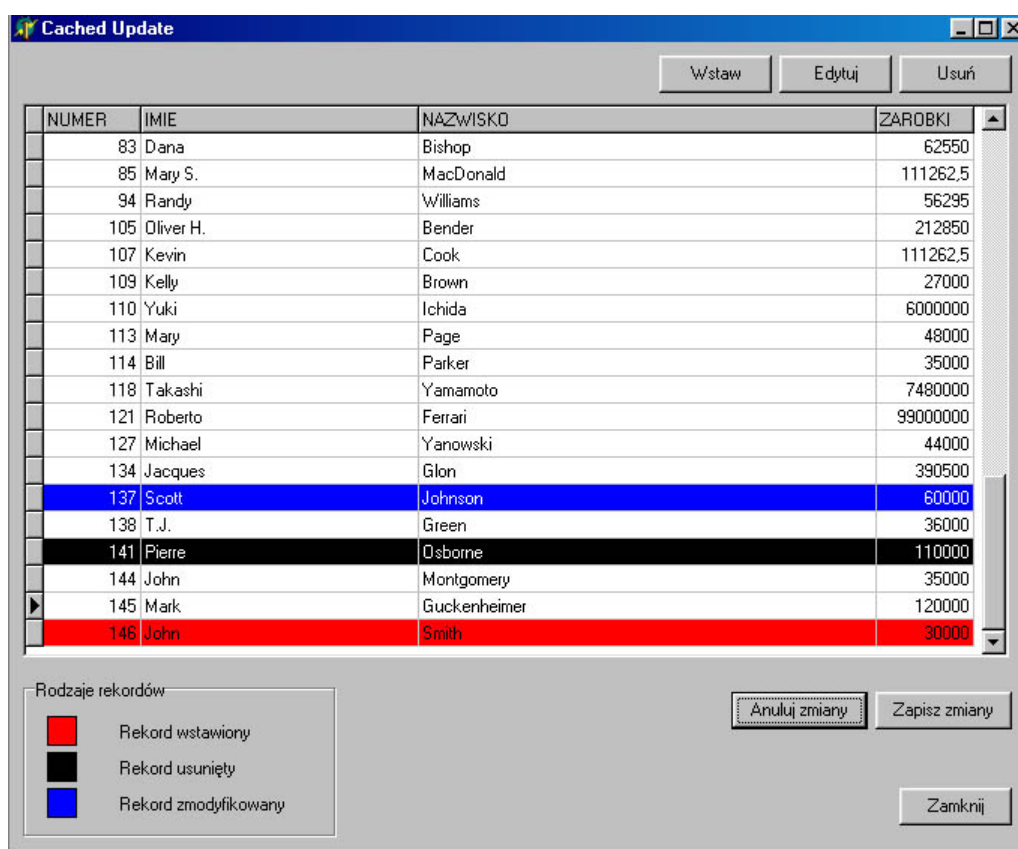
1) Zadanie: Poznanie właściwości CACHED UPDATE zbioru danych.

Buforowanie zmian (cached update) jest mechanizmem składającym się z trzech faz. Pierwsza z nich polega na pobraniu danych z serwera. Druga faza to lokalne edytowanie danych. Trzecia faza polega na jednoczesnym zatwierdzeniu wszystkich edytowanych danych.

Przed uruchomieniem Delphi warto stworzyć tabelę z przykładowymi danymi, na której będziemy wykonywali operacje usuwania, dodawania i edycji.

Program powinien działać w następujący sposób: Wczytujemy do DBGrid dane z nowo utworzonej tabeli (w naszym przypadku jest to tabela zawierająca kopię następujących danych: Emp\_No, First\_Name, Last\_Name, Salary z tablicy Employee). Następnie stosując przyciski: Wstaw, Edytuj, Usuń dokonujemy zmian, które jednak nie będą natychmiast zatwierdzone. Wszystkie zmienione rekordy powinny być odpowiednio oznaczone. Po dokonaniu zmian możemy je zatwierdzić (przycisk Zapisz zmiany) lub wycofać (przycisk Anuluj zmiany).

Okno programu może wyglądać następująco:



Tworząc program będziemy korzystać z następujących komponentów:

- a) Z zakładki InterBase
  - i) IBDatabase
  - ii) IBTransaction
  - iii) IBTable
- b) Z zakładki Data Access
  - i) DataSource
- c) Z zakładki Data Controls
  - i) DBGrid
- d) Z zakładki Standard
  - i) Button
  - ii) GroupBox
  - iii) Label
- e) Z zakładki Additional
  - i) Shape

- 2) Wrzucić na formatkę `IBDatabase1` i kliknąć dwa razy na tym komponencie. Wypełnić listę parametrów:  
Zaznaczyć `Local`  
Server  
Protocol  
Database: `C:/firebird/firebird2/examples/empbuild/employee.fdb`  
User Name: `SYSDBA`  
Password: `masterkey`  
Własność `LoginPrompt` ustawić na `FALSE`  
Własność `Connected` ustawić na `TRUE`

- 3) Wrzucić na formatkę `IBTransaction1`  
Dla własności `DefaultDatabase` wybrać `IBDatabase1`

- 4) Wrzucić na formatkę `IBTable1`  
Dla własności `Database` wybrać `IBDatabase1`  
Dla własności `Table Name` wybrać `Employee`  
Ustawić własność `Active` na `TRUE`  
Ustawić własność `Cached Update` na `TRUE`

- 5) Wrzucić na formatkę `DataSource1`  
Dla własności `DataSet` wybrać `IBTable1`

- 6) Wrzucić na formatkę `DBGrid1`  
Dla własności `DataSource` wybrać `DataSource1`  
W parametrze `Options` zaznaczyć `dgRowSelect` (dzięki temu `DBGrid` nie będzie edytowalny dopóki nie wciśniemy odpowiedniego przycisku)

- 7) Dla zdarzenia `FormActivate` powinny być wykonywane następujące akcje:

```
IBTable1.Open;  
IBTable1.FetchAll; //pobiera wszystkie rekordy (nie trzeba robić Last, First)  
IBTable1.UpdateRecordTypes:= IBTable1.UpdateRecordTypes + [cusDeleted];  
//dzięki tej zmianie będą pokazywane rekordy usunięte  
ButtonZapisz.Enabled:= False;  
ButtonAnuluj.Enabled:=False;
```

- 8) Procedure „rysująca” `DBGrid` (z uwzględnieniem kolorów zmienionych rekordów) może wyglądać następująco:

//korzystamy tu z właściwości zbioru danych `UpdateStatus`, która informuje jaka zmiana została dokonana dla konkretnego rekordu

Dla zdarzenia `OnDrawColumnCell` komponentu `DBGrid1`:

```
procedure TForm1.DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;  
  DataCol: Integer; Column: TColumn; State: TGridDrawState);  
begin  
  with TDBGrid(Sender).Canvas do  
    begin  
      case IBTable1.UpdateStatus of  
        usInserted : Brush.Color := clRed;  
        usModified :  
          begin  
            Brush.Color := clBlue;  
            Font.Color := clWhite;  
          end;  
        usDeleted :  
          begin  
            Brush.Color := clBlack;  
            Font.Color := clWhite;  
          end;  
      end;  
      TDBGrid(Sender).DefaultDrawColumnCell(Rect, DataCol, Column, State);  
    end;  
end;
```

```
end;
```

- 9) Do oprogramowania przycisków Wstaw, Edytuj, Usuń używamy metod odpowiednio: Insert, Edit, Delete. Dla zdarzeń OnClick dla przycisków Wstaw, Edytuj, Usuń wpisujemy odpowiednio kod:

```
IBTable1.Insert;  
IBTable1.Edit;  
IBTable1.Delete;
```

W celu umożliwienia edycji w **DBGrid**, czyli dla przycisków Wstaw i Edytuj musimy dodatkowo zmienić opcje komponentu **DBGrid**

```
DBGrid1.Options:= DBGrid1.Options - [dgRowSelect] + [dgEditing];
```

- 10) Po każdej zmianie pozycji w **DBGrid** należy znowu zablokować możliwość edycji (zdarzenie AfterScroll dla komponentu IBTable1):

```
procedure TForm1.IBTable1AfterScroll(DataSet: TDataSet);  
begin  
  DBGrid1.Options:= DBGrid1.Options + [dgRowSelect];  
  if IBTable1.UpdatesPending then  
  begin  
    ButtonZapisz.Enabled:=True;  
    ButtonAnuluj.Enabled:=True;  
  end;  
end;
```

Właściwość UpdatesPending sprawdza czy tryb buforowania zmian jest w trakcie działania, innymi słowy, czy są jeszcze jakieś niezatwierdzone zmiany.

- 11) Zatwierdzanie zmian:

```
procedure TForm1.ButtonZapiszClick(Sender: TObject);  
begin  
  IBTable1.UpdateRecordTypes:= IBTable1.UpdateRecordTypes - [cusDeleted];  
  with IBTable1 do  
  begin  
    if IBTransaction1.Active=False Then  
      IBTransaction1.StartTransaction;  
    try  
      ApplyUpdates;  
      IBTransaction1.Commit;  
      IBTable1.Open;  
    except  
      IBTransaction1.Rollback;  
    end;  
  end;  
  IBTable1.UpdateRecordTypes:= IBTable1.UpdateRecordTypes + [cusDeleted];  
  if not(IBTable1.UpdatesPending) then  
  begin  
    ButtonAnuluj.Enabled:=False;  
    ButtonZapisz.Enabled:=False;  
  end;  
end;
```

- 12) Wycofywanie zmian:

```
procedure TForm1.ButtonAnulujClick(Sender: TObject);  
begin  
  IBTable1.CancelUpdates;  
  if not(IBTable1.UpdatesPending) then  
  begin  
    ButtonAnuluj.Enabled:=False;  
    ButtonZapisz.Enabled:=False;  
  end;  
end;
```

- 13) Wrzuć na formatkę komponent GroupBox1. Ustaw własność Caption na Rodzaje rekordów.
- 14) Umieść na polu GroupBox1 anstępujące elementy:  
Shape1, Shape2, Shape3,  
Label1, Label2, Label3.  
Ustaw kolory kwadratów Shape - własność Brush - odpowiednio na  
clRed, clBlack, clBlue.  
Ustaw własności Caption dla komponentów Label odpowiednio jako  
Rekord wstawiony, Rekord usunięty, Rekord Zmodyfikowany.
- 15) Wrzuć na formatkę przycisk Button, ustaw Caption na Zamknij, zdarzenie OnClick niech wykonuje:  
Form1.Close;