

```

program Corbi;

uses
  Vcl.Forms,
  Console1 in 'Console1.pas' {Form2},
  Login1 in 'Login1.pas' {Form1},
  AnaAtl in 'AnaAtl.pas' {Form3},
  Vcl.Themes,
  Vcl.Styles,
  AnaDspl in 'AnaDspl.pas' {Form4},
  AnaGa in 'AnaGa.pas' {Form5},
  Gara in 'Gara.pas' {Form6},
  aiuto in 'aiuto.pas' {Form7},
  Classifiche in 'Classifiche.pas' {Form8},
  Tempi in 'Tempi.pas' {Form9},
  Dialogs,
  Windows,
  System.UITypes,
  Mancanti in 'Mancanti.pas' {Form10};

//no2
{$R *.res}
const
  TentaLancio='CodeCall.Net Mutex'; //no2
resourcestring //no2
  DUPLICATE_MSG='Il programma è stato già lanciato. Non consente duplicati'; //no2
var //no2
  vMutex : THandle; //no2
{$D+}
begin //codice per evitare doppia istanza
  vMutex := CreateMutex(nil, True, TentaLancio); //no2
  if GetLastError = ERROR_ALREADY_EXISTS then //no2
    begin //no2
      MessageDlg(DUPLICATE_MSG, mtError, [mbOK], 0); //no2
      Exit; //no2
    end; //no2
  try //no2
    Application.Initialize;
    Application.MainFormOnTaskbar := True;
    // Application.Title := 'Corbi';
    //Application.HelpFile := '.\corbi.chm';
    Application.CreateForm(TForm2, Form2);
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TForm7, Form7);
    Application.CreateForm(TForm9, Form9);
    Application.CreateForm(TForm10, Form10);
    Application.CreateForm(TForm8, Form8);
    Application.CreateForm(TForm6, Form6);
    Application.CreateForm(TForm3, Form3);
    Application.CreateForm(TForm4, Form4);
    Application.CreateForm(TForm5, Form5);

    //posporre a Form2 che contiene il database
    Application.ShowMainForm := False;
    Application.Run;
    Application.NormalizeAllTopMosts ;
  finally //no2

```

```
        CloseHandle (vMutex) ;  
    end;  
end.  
//no2  
//no2
```

```

unit Login1;    //19/01/2016

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, System.UITypes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
  Vcl.StdCtrls, Data.DB, ABSSMain, System.Types;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);

  private
    { Private declarations }
  public

    //class function Execute: Boolean;
  end;

var
  Form1: TForm1;
  dir, pwstr, cnvstr, pwtemp: string;           //pwstr è la password in ingresso:
  cnvstr è la pw convertita
  nuovapw: Boolean = false;
  cicla: boolean = False;
  // ATTENZIONE VALORE INSERITO AL DEPLOY DEL PROGRAMMA è identico alla pw del
  Database
  Jolly: string = 'Corbi77'; //PW dell'amministratore usata anche in Console1
implementation
  uses
    Console1 ;
  {$R *.dfm}
  {$D+}
  procedure TForm1.FormCreate(Sender: TObject); //preliminare login
  begin //verifica se c'è il db
    dir := GetCurrentDir + '\';
    pwstr := '';
    cnvstr := '';
    pwtemp := '';
    if FileExists(dir + 'corbi.ABS') = False then //se non c'è il database
    begin
      MessageDlg('ATTENZIONE - File Dati corrotto o inesistente' + #13 + ' Il
programma verrà chiuso ', mtError, [mbOK], 0 );
      Application.Terminate;
    end else
    begin // se c'è il database
      Form2.ABSTable1.TableName := 'Servizio';
      Form2.ABSTable1.Open;
      Form2.ABSTable1.First;
      Form2.ABSTable1.Edit;
      if (Form2.ABSTable1.FieldName('Password').IsNull) then //se non c'è la
pw CREARE LA PW
      begin

```

```

        ShowMessage('Assegnate una password: max 8 caratteri ');
        nuovapw:=True; //flag per nuova pw
        form1.Visible:=True;
    end
    else form1.Visible:=True; // continua se c'è la pw con la routine di
riconoscimento
    end;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char); // carico
login
var
n,L:integer;
p:Char;
riconosce:string;
begin
    if Key=#13 then // all'atto dell'entry della pw
    begin
        pwstr:=Edit1.Text; // pwstr è la password attualmente digitata
        if pwstr=Jolly then //Procedura per ripristinare la pw
        begin
            ShowMessage('ATTENZIONE - Ripristino password '+#13+'Il programma
verrà chiuso, rilancialo e'+#13
+'Digita una nuova password');
            Form2.ABSTable1.First; //Attenzione sembra che porre xx.first dopo edit
tolga dalla condizione di edit
            Form2.ABSTable1.Edit;
            Form2.ABSTable1.FieldName('password').AsString:=''; //cancella la
vecchia password per ripristinarla
            Form2.ABSTable1.Post;
            Application.Terminate;
        end;
        L:= Length(pwstr);
        if L=0 then
        begin
            ShowMessage('Attenzione - Non è stata inserita alcuna password');
            Exit;
        end;
        if L>8 then //verifica la lunghezza massima
        begin
            ShowMessage('Numero di caratteri superiore ad 8, ripeti');
            Edit1.Text:='';
            Exit;
        end;
        if (nuovapw=True) and (cicla=False) then // Se è nuova password
        begin
            cnvstr:='';
            pwtemp:=pwstr; //si pone la pw nella memoria temporanea
            Edit1.Text:=''; //verifica pw
            ShowMessage('Conferma la password inserita');
            cicla:=True;
            Exit;
        end;
        if (nuovapw=True) and (cicla=True) then //sempre la nuova nel secondo
ciclo
        begin
            if pwtemp=pwstr then
            begin

```

```

        for n := 1 to L do      //procedura di cryptaggio
        begin
            p:= chr(ord(pwstr[n])+(L+1-n));
            cnvstr:=cnvstr + p;
        end;
        Form2.ABSTable1.Open;
        if Form2.ABSTable1.RecordCount=0 then
        Form2.ABSTable1.Insert else
        Form2.ABSTable1.First;
        Form2.ABSTable1.Edit;
        Form2.ABSTable1.FieldName('password').AsString:=cnvstr;
        Form2.ABSTable1.Post;
        Form2.ABSTable1.Close;
        form1.Visible:=False;
        cicla:=False;
        pwtemp:='';
        pwstr:='';
        nuovapw:=False;
        ShowMessage('La password è stata regolarmente
registrata'+#13+'Memorizzala in luogo sicuro');
        Form1.Visible:=False;
        Form1.Close;
        Form2.Visible:=True;
    end else
    begin
        MessageDlg('ATTENZIONE - I due inserimenti non
coincidono'+#13+'Ripeti con attenzione', mtError, [mbOK], 0 );
        Edit1.Text:='';
        cicla:=False;
        pwtemp:='';
        pwstr:='';
        Exit;
    end;
end
else      //la pw non è nuova quindi PROCEDURA DI RICONOSCIMENTO
begin
    for n := 1 to L do      //procedura di cryptaggio
    begin
        p:= chr(ord(pwstr[n])+(L+1-n));
        cnvstr:=cnvstr + p;
    end;
    Form2.ABSTable1.Open;
    Form2.ABSTable1.First;
    riconosce:=Form2.ABSTable1.FieldName('password').AsString;
    if cnvstr=ricosce then
    begin
        ShowMessage('Accesso consentito') ;      // ACCESSO
        Form2.ABSTable1.Open;
        Form2.ABSTable1.INSERT;
        Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
        Form2.ABSTable1.FieldName('what').AsString:='Accesso alla
Console';
        Form2.ABSTable1.FieldName('OUT').AsString:='';
        Form2.ABSTable1.Post;
        Form2.ABSTable1.Close;
        Form1.Visible:=False ;
        Form1.Close;
        Form2.Visible:=True;
    end;
end;

```

```

        end
    else //La password inserita è sbagliata
    begin
        if cicla=True then //ha sbagliato per la seconda volta
        begin
            if MessageDlg('Password errata ! Vuoi ripristinarla ?',
                mtConfirmation, [mbYes, mbNo], 0, mbYes) = mrYes then
            begin
                ShowMessage('Digita la password originale di
registrazione');

                Edit1.Text:='';
                Exit;
            end else Application.terminate;
            end
        else
        begin
            MessageDlg('ATTENZIONE - Accesso negato', mtError, [mbOK], 0 );
            pwstr:='';
            cnvstr:='';
            edit1.text:='';
            cicla:=True ;
            Exit;
        end;
    end;

    end;

end;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action:=caFree;
end;

end.

```

```
unit Console1;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,  
System.Classes, Vcl.Graphics,  
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Data.DB, ABSMain, frxClass,  
frxDBSet, System.Types, JvBaseDlg, JvSHFileOperation, frxExportPDF,  
Vcl.ExtCtrls, System.UITypes,  
Vcl.Imaging.jpeg, Winapi.ShellAPI, Vcl.HtmlHelpViewer,  
Vcl.Buttons, Vcl.AppEvnts;
```

```
type
```

```
TForm2 = class (TForm)  
Esci: TButton;  
Button1: TButton;  
Button2: TButton;  
Button3: TButton;  
Button4: TButton;  
ABSDatabasel: TABSDatabase;  
ABSTable1: TABSTable;  
ABSTable3: TABSTable;  
ABSTable3D: TABSTable;  
frxDBDataset1: TfrxDBDataset;  
ABSTable2: TABSTable;  
ABSTable4: TABSTable;  
ABSTable5: TABSTable;  
ABSTable6: TABSTable;  
ABSTable9: TABSTable;  
ABSTable7: TABSTable;  
ABSTable8: TABSTable;  
ABSTable12: TABSTable;  
ABSTable10: TABSTable;  
ABSTable11: TABSTable;  
Button5: TButton;  
Button6: TButton;  
Button7: TButton;  
frxPDFExport1: TfrxPDFExport;  
ABSTable16T: TABSTable;  
Button8: TButton;  
Button9: TButton;  
ABSQuery2: TABSQuery;  
frxReport2: TfrxReport;  
frxDBDataset2: TfrxDBDataset;  
Button10: TButton;  
Button11: TButton;  
frxReportLog: TfrxReport;  
frxDBDatasetLog: TfrxDBDataset;  
Label1: TLabel;  
JvSHFile1: TJvSHFileOperation;  
frxReport1: TfrxReport;  
BitBtn1: TBitBtn;  
Image1: TImage;  
ApplicationEvents1: TApplicationEvents;  
procedure FormCreate(Sender: TObject);  
procedure EsciClick(Sender: TObject);  
procedure Button1Click(Sender: TObject);
```

```

procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure JVSHFile1Show(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure ApplicationEvents1ShortCut(var Msg: TWMKey; var Handled: Boolean);

private
    { Private declarations }
public
    function GetAppVersionStr: string;
    { Public declarations }
end;

var
    Form2: TForm2;
    consflg:Boolean=True;    // quando èTrue Form2 èclosed
    Log1,versione:string;
    riparato:Boolean;
    //KBHook: HHook;
    //function KeyboardHookProc(Code: Integer; WordParam: Word; LongParam:
    LongInt): LongInt; stdcall;
implementation
    uses        //per utilizzare il form3 e Form4 ecc
        AnaAtl,Login1,AnaDspl,AnaGa,Gara,Classifiche,aiuto;

    {$R *.dfm}

procedure TForm2.FormCreate(Sender: TObject);
begin

    ABSDatabase1.DatabaseFileName:=ExtractFilePath(Application.ExeName)+'corbi.ABS
    ';
    //Application.HelpFile := ExtractFilePath(Application.ExeName) + 'Corbi.chm';
    Form2.ABSDatabase1.Open;
    Form2.ABSTable1.TableName:='Servizio';
    Form2.ABSTable1.Open;
    Form2.ABSTable1.Edit;
    //registra su Servizio
    Form2.ABSTable1.INSERT;
    Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
    Form2.ABSTable1.FieldName('what').AsString:='Accesso alla Consolle';
    Form2.ABSTable1.FieldName('OUT').AsString:='';
    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
    Form2.ABSTable6.TableName:='Base';
    Form2.ABSTable4.TableName:='AnaDspl';
    Form2.ABSTable8.TableName:='GRPA';
    Form2.ABSTable9.TableName:='GRPB';
    Form2.ABSTable5.TableName:='AnaGa';

```



```

Form2.ABSTable7.TableName:='Gara';
Form2.ABSTable2.TableName:='AnaAtl';
Form2.ABSTable3.TableName:='SelezioneGare';
Form2.ABSTable3D.TableName:='SelezioneDSPL';
Form2.ABSTable10.TableName:='repBase';
Form2.ABSTable11.TableName:='repGRPA';
Form2.ABSTable12.TableName:='repGRPB';
Form2.ABSTable16T.TableName:='T16T';
Form2.Visible:=False;
consflg:=False;           //per associare l'hook alla pressione di un tasto

//KBHook:=SetWindowsHookEx(WH_KEYBOARD,@KeyboardHookProc,HInstance,GetCurrentTh
hreadId());
  GetAppVersionStr;
  caption:=caption + ' - ver: '+versione;
end;

{ function KeyboardHookProc(Code: Integer; WordParam: Word; LongParam: LongInt)
: LongInt;
  begin
    //per chiamare l'help da F1
    if WordParam=VK_F1 then
      ShellExecute(0,'open',PChar('Corbi.chm'), nil, nil, SW_SHOWNormal)
    else
      Result:=0; //necessaria per sbloccare l'intercettazione per tasti diversi da
F1
  end; }

{ //forma ridotta
function TForm2.GetAppVersionStr: string;
var
  Rec: LongRec;
begin
  Rec := LongRec(GetFileVersion(ParamStr(0)));
  Result := Format('%d.%d', [Rec.Hi, Rec.Lo]);
  versione:=Result
end;
}

function TForm2.GetAppVersionStr: string;
var
Exe: string;
Size, Handle: DWORD;
Buffer: TBytes;
FixedPtr: PVSFixedFileInfo;
begin
  Exe := ParamStr(0);
  Size := GetFileVersionInfoSize(PChar(Exe), Handle);
  if Size = 0 then
    RaiseLastOSError;
  SetLength(Buffer, Size);
  if not GetFileVersionInfo(PChar(Exe), Handle, Size, Buffer) then
    RaiseLastOSError;
  if not VerQueryValue(Buffer, '\\', Pointer(FixedPtr), Size) then
    RaiseLastOSError;
  Result := Format('%d.%d.%d.%d',
[LongRec(FixedPtr.dwFileVersionMS).Hi, //major
  LongRec(FixedPtr.dwFileVersionMS).Lo, //minor
  LongRec(FixedPtr.dwFileVersionLS).Hi, //release

```

```

    LongRec(FixedPtr.dwFileVersionLS).Lo]); //build
    versione:= Result;
end;

procedure TForm2.JVSHFile1Show(Sender: TObject);
begin
    ShowMessage('Effettuato il backup del Database');
end;

procedure TForm2.Button1Click(Sender: TObject); //Anagrafica atleti
begin //Form2 =Console
    if Form2.visible=True then
        Form2.Visible:=False;
    if atlflg=False then
        begin
            Form3:= TForm3.Create(Self);
            Form3.show ;
        end
    else //essendo atlflg True occorre ri-creare il Form3 prima di poterlo
        begin // rendere visibile se si ripete il click
            Form3:= TForm3.Create(Self);
            Form3.Visible:=True;
            atlflg:=False;
        end;
end;

procedure TForm2.Button2Click(Sender: TObject); //Anagrafiva discipline
begin
    if Form2.visible=True then
        Form2.Visible:=False;
    if Dsplflg=False then
        begin
            Form4:= TForm4.Create(Self);
            Form4.show ;
        end
    else //essendo atlflg True occorre ri-creare il Form3 prima di poterlo
        begin // rendere visibile se si ripete il click
            Form4:= TForm4.Create(Self);
            Form4.Visible:=True;
        end;
end;

procedure TForm2.Button3Click(Sender: TObject); //Anagrafica Gare
begin
    if Form2.visible=True then
        Form2.Visible:=False;
    if AnaGAFlg=False then
        begin
            Form5:= TForm5.Create(Self);
            Form5.show;
        end
    else //essendo AnaGAFlg True occorre ri-creare il Form5 prima di
        begin // rendere visibile se si ripete il click
            Form5:= TForm5.Create(Self);
            Form5.Visible:=True;
        end;
end;

```

```

        end;
end;

procedure TForm2.Button4Click(Sender: TObject); // Gara Corrente Iscrizioni
begin
    if Form2.visible=True then
        Form2.Visible:=False;
        if GAFlg=False then
            begin
                Form6:= TForm6.Create(Self);
                Form6.show ;
            end
        else //essendo GAFlg True (cioèForm chiuso) occorre ri-creare il Form6
            begin //prima di poterlo rendere visibile se si ripete il click
                Form6:= TForm6.Create(Self);
                Form6.Visible:=True;
            end;
end;

end;

procedure TForm2.Button5Click(Sender: TObject); // Backup
begin
    if MessageDlg('Confermi di voler effettuare il backup del DataBase ?',
        mtconfirmation, [mbYes,mbNo], 0) = mrYes then
        begin
            ABSDatabase1.Close;
            JVSHFile1.Execute;
            // JVSHFile1.OnClose(Self);
            Form2.ABSTable1.Open;
            Form2.ABSTable1.Last;
            if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
                Form2.ABSTable1.Insert;
            Form2.ABSTable1.Edit;
            Form2.ABSTable1.FieldName('IN').AsDateTime:=NOW;
            Form2.ABSTable1.FieldName('what').AsString:='Effettuato Backup';
            Form2.ABSTable1.Post;
            Form2.ABSTable1.Close;
        end else Exit;

end;

end;

procedure TForm2.Button6Click(Sender: TObject); // Ripara
    Var
        result:string;
begin
    Form2.ABSDatabase1.Close;
    result:= ABSDatabase1.RepairDatabase;
    if result = ' ' then
        ShowMessage('Il Database è stato verificato ')else
        ShowMessage('Il Database è corrotto - Usa il backup');

end;

end;

procedure TForm2.Button7Click(Sender: TObject); //Classifiche
begin
    if Form2.visible=True then
        Form2.Visible:=False;
        if ClassFlg=False then

```

```

begin
    Form8:= TForm8.Create(Self);
    Form8.show ;
end
else      //essendo ClassFlg True occorre ri-creare il Form8 prima di
poterlo
begin      // rendere visibile se si ripete il click
    Form8:= TForm8.Create(Self);
    Form8.Visible:=True;
end;
end;

procedure TForm2.Button8Click(Sender: TObject);    //Azzerata Storico
begin
    if MessageDlg('Stai per azzerare l''intero Storico delle Classifiche. Vuoi
continuare ?',
        mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
        begin
            Form2.ABSTable16T.Close;
            Form2.ABSTable16T.EmptyTable;
            ShowMessage('Tutta lo Storico è stato cancellato !');
            Form2.ABSTable1.Open;
            Form2.ABSTable1.Last;
            Form2.ABSTable1.Insert;
            Form2.ABSTable1.Edit;
            Form2.ABSTable1.FieldName('IN').AsDateTime:=NOW;
            Form2.ABSTable1.FieldName('what').AsString:='Cancellato lo
Storico delle Classifiche';
            Form2.ABSTable1.Post;
            Form2.ABSTable1.Close;
        end;
    end;

end;

procedure TForm2.Button9Click(Sender: TObject);    //Complessivo
var
    ca1,ca2 : Integer;
    d1,d2:TDateTime;
    trovato:Boolean;
begin
    ca2:=0;
    ABSTable16T.Open;
    Try
        if MessageDlg('Desideri ottenere il punteggio storico per Atleta ?'
            +#13+'Dovrai fornire i codici della prima ed ultima Gara da conteggiare',
            mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrNo then Exit;
        ca1:=StrToInt(InputBox('Codice Gara iniziale','Inserisci la prima',''));
        if ca1<>0 then
            ca2:=StrToInt(InputBox('Codice Gara finale','Inserisci l''ultima',''));
    Except
        ShowMessage('Errore nell''inserimento del dato');
        Exit;
    End;
    trovato:=ABSTable16T.Locate('CODGA',ca1,[]);
    if trovato=True then
        d1:=ABSTable16T.FieldName('DATAGA').AsDateTime else
        begin
            ShowMessage('Gara codice '+inttostr(ca1)+' non esiste.');
```

```

        Exit;
    end;
    trovato:=ABSTable16T.Locate('CODGA',ca2,[]);
    if trovato=True then
        d2:=ABSTable16T.FieldByName('DATAGA').AsDateTime else
        begin
            ShowMessage('La Gara con codice '+IntToStr(ca2)+' non contiene dati di
classifiche.');
```

Exit;

```

        end;

    Form8.ABSTable19.Close;           //T19 = totale
    Form8.ABSTable19.EmptyTable;
    Form8.ABSTable19.Open;
    with ABSQuery2 do
        begin
            Close;
            SQL.Text:='INSERT INTO totale(CODATL,COGNOME,NOME,NAZIONE,RNK) SELECT
CODATL,COGNOME,NOME,NAZIONE,SUM(RNK) AS RNK ' +
            ' FROM T16T where CODGA BETWEEN '+QuotedStr(IntToStr(ca1))+' AND
'+QuotedStr(IntToStr(ca2))+
            ' GROUP BY CODATL,COGNOME,NOME,NAZIONE ORDER BY RNK desc';
            ExecSQL;
            //Open;
        end;
        frxReport2.Variables['ini']:= quotedstr(DateToStr(d1));
        frxReport2.Variables['fin']:= quotedstr(DateToStr(d2));
        frxReport2.ShowReport;
        ABSTable16T.Close;
        Form8.ABSTable19.Close;
        Form8.ABSTable19.EmptyTable;

end;

procedure TForm2.Button10Click(Sender: TObject);
var
    s:string;
begin
    //per mascherare i caratteri premetter #31

    s:=InputBox('Richiesta PW',#31'Inserisci PW dell''Amministratore','');
    if s<>'Corbi77' then Exit;
    frxReportLog.ShowReport;
end;

procedure TForm2.Button11Click(Sender: TObject); //Cancella il log
var
    s:string;
    n:Integer;
begin
    s:=InputBox('Richiesta PW',#31'Inserisci PW dell''Amministratore','');
    Form2.ABSTable1.Open;
    Form2.ABSTable1.Edit;
    if s<>'Corbi77' then Exit;
    n:=Form2.ABSTable1.RecordCount;
    if MessageDlg('Il LOG ha '+IntToStr(n)+' records, vuoi cancellarlo ? ',
mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrNo then
        Exit;
    Form2.ABSTable1.First;

```

```

while not Form2.ABSTable1.Eof do
begin
    if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
        Form2.ABSTable1.Next else
        Form2.ABSTable1.Delete; //NB il delete causa giàun next
    end;
    Form2.ABSTable1.Close;
end;

procedure TForm2.EsciClick(Sender: TObject); //Esci
begin
    Form2.ABSTable1.Open;
    Form2.ABSTable1.Last;
    Form2.ABSTable1.Insert;
    Form2.ABSTable1.Edit;
    Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
    Form2.ABSTable1.FieldName('what').AsString:='Chiusura della Sessione';
    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
    ABSDataBasel.Close;
    Form2.release;
    consflg:=True;
    Application.Terminate;
end;

//tramite l'uso del componente applicationEvent si chiama l'help
//mettere negli Uses HTMLHelpViewer . Non serve altro
procedure TForm2.ApplicationEvents1ShortCut(var Msg: TWMKey;
var Handled: Boolean);
begin
    if (Msg.CharCode = VK_F1) then
        begin
            ShellExecute(0, 'open', PChar('Corbi.chm'), nil, nil, SW_SHOWNormal);
            Handled := True;
        end;
end;

procedure TForm2.BitBtn1Click(Sender: TObject); //RESET del Database
begin
    if MessageDlg('Stai per azzerare l''intero Database. Vuoi continuare ?',
        mtWarning, [mbYes, mbNo], 0, mbNo) = mrYes then
        begin
            if MessageDlg('Questa operazione non è ripristinabile, '+#13+
                'Tutti gli archivi verranno cancellati !'+#13+' Vuoi continuare ?',
                mtWarning, [mbYes, mbNo], 0, mbNo) = mrYes then
                begin
                    Form2.ABSTable1.Close;
                    Form2.ABSTable1.EmptyTable;
                    Form2.ABSTable2.Close;
                    Form2.ABSTable2.EmptyTable;
                    Form2.ABSTable3.Close;
                    Form2.ABSTable3.EmptyTable;
                    Form2.ABSTable4.Close;
                    Form2.ABSTable4.EmptyTable;
                    Form2.ABSTable5.Close;
                    Form2.ABSTable5.EmptyTable;
                    Form2.ABSTable6.Close;
                    Form2.ABSTable6.EmptyTable;
                end;
            end;
        end;
end;

```

```

Form2.ABSTable7.Close;
Form2.ABSTable7.EmptyTable;
Form2.ABSTable8.Close;
Form2.ABSTable8.EmptyTable;
Form2.ABSTable9.Close;
Form2.ABSTable9.EmptyTable;
Form2.ABSTable10.Close;
Form2.ABSTable10.EmptyTable;
Form2.ABSTable11.Close;
Form2.ABSTable11.EmptyTable;
Form2.ABSTable12.Close;
Form2.ABSTable12.EmptyTable;
Form6.ABSTable13.Close;
Form6.ABSTable13.EmptyTable;
Form6.ABSTable14.Close;
Form6.ABSTable14.EmptyTable;
Form8.ABSTable16.Close;
Form8.ABSTable16.EmptyTable;
Form8.ABSTable18.Close;
Form8.ABSTable18.EmptyTable;
Form2.ABSTable16T.Close;
Form2.ABSTable16T.EmptyTable;
Form8.ABSTable16R.Close;
Form8.ABSTable16R.EmptyTable;
Form8.ABSTable18.Close;
Form8.ABSTable18.EmptyTable;
Form8.ABSTable19.Close;
Form8.ABSTable19.EmptyTable;
Form2.ABSTable3D.Close;
Form2.ABSTable3D.EmptyTable;
ShowMessage('Tutti gli archivi sono stati azzerati.');
```

end else Exit;

end else Exit;

end;

end.

```

unit AnaAtl;    // Unità per il carico dell'anagrafe atleti

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, ABSSMain, Data.DB,
  Vcl.Grids, Vcl.DBGrids, Vcl.DBCGrids, Vcl.ExtCtrls,
  Vcl.DBCtrls, System.UITypes,
  frxClass, frxDBSet, System.Types;

type
  TForm3 = class(TForm)
    Button1: TButton;
    ABSQuery1: TABSQuery;
    dbgrd1: TDBGrid;
    dbnvgr1: TDBNavigator;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Button5: TButton;
    ds1: TDataSource;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure dbnvgr1Click(Sender: TObject; Button: TNavigateBtn);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure ds1DataChange(Sender: TObject; Field: TField);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
  atlflg: Boolean = True; // Quando è True il form3 è closed
  filtro: string; // per SQL
  inserisco: Boolean = False; // flag per inserimento dati;
  canclose: Boolean = True;

implementation
  uses
    Console1;
  {$R *.dfm}
  {$D+}
  procedure TForm3.FormCreate(Sender: TObject);
begin
  // ricordarsi di assegnare una datasource al Navigator
  // Form2.ABSTable2.TableName:='AnaAtl';

```



```

Form2.ABSTable2.Open;
atlflg:=False;
ds1.DataSet:=Form2.ABSTable2;//aggiunto per evitare circular DataLink error
Form2.ABSTable2.Edit;
Form2.ABSTable1.Open;
Form2.ABSTable1.INSERT;
Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
Form2.ABSTable1.FieldName('what').AsString:='Accesso alla Anagrafica
Atleti';
Form2.ABSTable1.FieldName('OUT').AsString:='';
Form2.ABSTable1.Post;
Form2.ABSTable1.Close;
end;

//Intercetto DBNavigator
procedure TForm3.dbnvgr1Click(Sender: TObject; Button: TNavigateBtn);

begin    //per selezionare il primo campo

    case button of
        nbInsert:    //se inserisce un nuovo record
            begin
                Form2.ABSTable2.Last;
                Form2.ABSTable2.Append;
                inserisco:=True; //inserisce il flag nel nuovo record
                dbgrd1.SetFocus;
                DBGrd1.SelectedField := Form2.ABSTable2.FieldName('NOME');
            end ;

        nbPost:      //se lo carica nella tabella
            inserisco:=False ;
    end;
end;

    procedure TForm3.ds1DataChange(Sender: TObject; Field: TField);
begin

end;

// per monitorare il form close quando si èaggiunto un record
procedure TForm3.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    if inserisco=True then
        if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0)
        = mrYes
        then
            begin
                CanClose:=False;
                Exit;
            end
        else
            begin
                CanClose:=True;
                if inserisco=True then
                    inserisco:=False;
                    Button1Click(Self);
                end;
            if CanClose=False then

```

```

        CanClose:=True;
end;

procedure TForm3.Button1Click(Sender: TObject); // Esci
begin
    if inserisco=True then //se ha inserito un record
        begin
            FormCloseQuery(Self,canclose);
        end
    else
        begin
            Form2.ABSTable1.Open;
            Form2.ABSTable1.Last;
            if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
                Form2.ABSTable1.Insert;
            Form2.ABSTable1.Edit;
            Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
            Form2.ABSTable1.Post;
            Form2.ABSTable1.Close;
            Form2.ABSTable2.Close;
            Form3.Close; // NB chiama il can close
            atlflg:=True; //per consentire di monitorizzare che Form3 è closed
            Form2.Visible:=True;
            Form3.Release;
        end;
end;

procedure TForm3.Button2Click(Sender: TObject); //cerca
label
vai;
var
chi:array[1..2]of shortint;
i:ShortInt;
begin
    ds1.DataSet:=ABSQuery1;
    i:=0; //determina se nome o/e cognome è in ricerca
    if edit1.text<>' ' then
        begin
            i:=i+1;
            chi[i]:=1;
        end;
    if edit2.Text<>' ' then
        begin
            i:=i+1;
            chi[i]:=2;
        end;
    if chi[2]>0 then //elabora i query
        begin
            filtro:='NOME LIKE ' + QuotedStr('%'+Edit1.Text+'%')+' AND COGNOME LIKE '
            +quotedStr('%'+edit2.text+'%');
            goto vai;
        end;
    if chi[1]>0 then
        begin
            if (edit1.Text<>' ') then
                filtro:= 'NOME LIKE '+quotedstr('%'+Edit1.Text+'%');

```

```

        if (edit2.Text<>'') then
            filtro:='COGNOME LIKE '+QuotedStr('%'+Edit2.Text+'%');
            goto vai;
        end;
    vai :
        //effettua il query
        with ABSQuery1 do
            begin
                close;
                if i>0 then
                    begin
                        SQL.Text := 'select * from AnaAtl where '+filtro; //Attenzione agli spazi
prima e dopo gli apici
                        ExecSql;
                    end else
                    begin
                        ShowMessage('Nessun elemento di ricerca inserito');
                        Exit;
                    end;
                Open;
            end;
        end;

procedure TForm3.Button3Click(Sender: TObject); //Resetta la ricerca
begin
    Edit1.Text:='';
    Edit2.Text:='';
    ds1.DataSet:=Form2.ABSTable2;
end;

        //STAMPA
        procedure TForm3.Button4Click(Sender: TObject);
begin
    Form2.frxReport1.ShowReport;
    if MessageDlg('Vuoi stamparlo ?' , mtconfirmation, [mbYes,mbNo], 0,mbNo) =
mrNo
        then Exit else
            Form2.frxReport1.Print;
end;

//AZZERARE il database Atleti
procedure TForm3.Button5Click(Sender: TObject);
begin
    if MessageDlg('Stai per azzerare l''intera Anagrafe degli atleti. Vuoi
continuare ?',
        mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
        begin
            Form2.ABSTable2.Close;
            Form2.ABSTable2.EmptyTable;
            ShowMessage('Tutta l''anagrafe è stata cancellata !');
        end;
end;

end;

end.

```

```

unit AnaDspl;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, ABSSMain, Vcl.ExtCtrls,
    Vcl.DBCtrls, Vcl.StdCtrls, Vcl.Grids, Vcl.DBGrids, frxClass,
    frxDBSet, System.UITypes,
    frxExportPDF;

type
    TForm4 = class (TForm)
        dbgrd4: TDBGrid;
        Label1: TLabel;
        Edit1: TEdit;
        Button1: TButton;
        Button2: TButton;
        Button3: TButton;
        Button4: TButton;
        Button5: TButton;
        dbnvgr4: TDBNavigator;
        ds4: TDataSource;
        ABSQuery4: TABSQuery;
        frxReport4: TfrxReport;
        frxDBDataset4: TfrxDBDataset;
        frxPDFExport4: TfrxPDFExport;
        Label2: TLabel;
        procedure Button5Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button4Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure dbnvgr4Click(Sender: TObject; Button: TNavigateBtn);
        procedure dbgrd4ColExit(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form4: TForm4;
    DsplFlg: Boolean = True;      // il crea lo cambia a False
    inserisco4: Boolean = False;
    CanClose: Boolean = True;
    filtro: string;

implementation
    uses
        Console1;
    {$R *.dfm}
    {$D+}
    procedure TForm4.FormCreate(Sender: TObject);
begin
    Form2.ABSTable4.TableName := 'AnaDspl';

```

```

Form2.ABSTable4.Open;
FormatDateTime('nn:ss,zzz',Form2.ABSTable4.FieldName('TMAX').asDateTime)
;

//TDateTimeField(Form2.ABSTable4.FieldName('TMAX')).DisplayFormat:='nn:ss,zz
z';
DsplFlg:=False;
ds4.DataSet:=Form2.ABSTable4; //aggiunto per evitare circular DataLink error
Form2.ABSTable4.Edit;
Form2.ABSTable1.Open;
Form2.ABSTable1.INSERT;
Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
Form2.ABSTable1.FieldName('what').AsString:='Accesso alla Anagrafica
Discipline';
Form2.ABSTable1.FieldName('OUT').AsString:='';
Form2.ABSTable1.Post;
Form2.ABSTable1.Close;
end;

procedure TForm4.Button1Click(Sender: TObject); //Cerca
begin
    ds4.DataSet:=ABSQuery4;
    if Edit1.Text<>' ' then
    begin
        filtro:='DESCR LIKE ' + QuotedStr('%'+Edit1.Text+'%');
        //effettua il query
        with ABSQuery4 do
        begin
            close;
            SQL.text := 'select * from AnaDspl where '+filtro; //Attenzione agli
spazi prima e dopo gli apici
            ExecSql;
            open;
        end;
    end else
    begin
        ShowMessage('Nessun elemento di ricerca inserito');
        Exit;
    end;
end;

procedure TForm4.Button2Click(Sender: TObject); //Annulla la ricerca
begin
    Edit1.Text:='';
    ds4.DataSet:=Form2.ABSTable4;
end;

procedure TForm4.Button3Click(Sender: TObject); //Stampa
begin
    frxReport4.ShowReport; //??
    if MessageDlg('Vuoi stamparlo ?' , mtconfirmation, [mbYes,mbNo], 0,mbNo) =
mrNo
    then Exit else
        frxReport4.Print;
end;

```

```

procedure TForm4.Button4Click(Sender: TObject);    //Azzerare la tabella
begin
    if MessageDlg('Stai per azzerare l''intera Anagrafe delle Discipline. Vuoi
    continuare ?',
    mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
        begin
            Form2.ABSTable4.Close;
            Form2.ABSTable4.EmptyTable;
            ShowMessage('Tutta l''anagrafe è stata cancellata !');
        end;
end;

procedure TForm4.Button5Click(Sender: TObject);    //Esci
begin
    if inserisco4=True then //se ha inserito un record
        begin
            FormCloseQuery(Self, cancel);
        end
    else
        begin
            Form2.ABSTable1.Open;
            Form2.ABSTable1.Last;
            if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
                Form2.ABSTable1.Insert;
                Form2.ABSTable1.Edit;
                Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
                Form2.ABSTable1.Post;
                Form2.ABSTable1.Close;
            Form2.ABSTable4.Close;
            Form4.Close;    // NB chiama il can close
            DsplFlg:=True;    //per consentire di monitorizzare che Form3 è closed
            Form2.Visible:=True;
            Form4.Release;
        end;
    end;

procedure TForm4.dbgrd4ColExit(Sender: TObject);
begin
    FormatDateTime('nn:ss,zzz',Form2.ABSTable4.FieldName('TMAX').asDateTime) ;

    //TDateTimeField(Form2.ABSTable4.FieldName('TMAX')).DisplayFormat:='nn:ss,zzz';
end;

procedure TForm4.dbnvgr4Click(Sender: TObject; Button: TNavigateBtn);
begin
    // azioni del DBNavigator
    case button of
        nbInsert:    //se inserisce un nuovo record
            begin
                Form2.ABSTable4.Last;
                Form2.ABSTable4.Append;
                inserisco4:=True; //inserisce il flag nel nuovo record
                dbgrd4.SetFocus;
                DBGrd4.SelectedField := Form2.ABSTable4.FieldName('DESCR');
            end ;

        nbPost:    //se lo carica nella tabella

```

```

    begin
        inserisco4:=False ;

FormatDateTime('nn:ss,zzz',Form2.ABTable4.FieldByName('TMAX').asDateTime) ;

//TDateTimeField(Form2.ABTable4.FieldByName('TMAX')).DisplayFormat:='nn:ss,zz
z';
    end;

end;
end;

procedure TForm4.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    // Chiusura del Form 4 DSPL
    if inserisco4=True then
        if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) =
mrYes
            then
                begin
                    CanClose:=False;
                    Exit;
                end
            else
                begin
                    CanClose:=True;
                    if inserisco4=True then
                        inserisco4:=False;
                        Button5Click(Self);
                    end;
                end;
            end;
end;
end.

```

```
unit AnaGa;                                //Anagrafica Gare    11/04/2015
```

interface

uses

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,  
System.Classes, Vcl.Graphics,  
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids,  
Vcl.DBGrids, System.UITypes,  
Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.DBCtrls, ABSMain, frxClass, frxDBSet,  
frxExportPDF;
```

type

```
TForm5 = class (TForm)  
    dbgrd51: TDBGrid;  
    ds51: TDataSource;  
    dbgrd52: TDBGrid;  
    ds52: TDataSource;  
    Button1: TButton;  
    dbnvgr51: TDBNavigator;  
    dbgrd55: TDBGrid;  
    ds55: TDataSource;  
    dbnvgr55: TDBNavigator;  
    cbb55: TComboBox;  
    ABSQuery6: TABSQuery;  
    Label1: TLabel;  
    Label2: TLabel;  
    Label3: TLabel;  
    Edit1: TEdit;  
    Edit2: TEdit;  
    Button2: TButton;  
    Label4: TLabel;  
    Button3: TButton;  
    Button4: TButton;  
    shp1: TShape;  
    Label5: TLabel;  
    Label6: TLabel;  
    Edit3: TEdit;  
    Button5: TButton;  
    Button6: TButton;  
    frxReport51: TfrxReport;  
    frxDBDataset51: TfrxDBDataset;  
    Button7: TButton;  
    frxPDFExport51: TfrxPDFExport;  
    Button8: TButton;  
    Button9: TButton;  
    Button10: TButton;  
    shp2: TShape;  
    Label7: TLabel;  
    frxReport5: TfrxReport;  
    frxDBDatasetBA: TfrxDBDataset;  
    frxDBDatasetGA: TfrxDBDataset;  
    frxDBDatasetGB: TfrxDBDataset;  
    procedure FormCreate(Sender: TObject);  
    procedure Button1Click(Sender: TObject);  
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);  
    procedure cbb55Change(Sender: TObject);  
    procedure dbgrd52MouseUp(Sender: TObject; Button: TMouseButton;
```



```

    Shift: TShiftState; X, Y: Integer);
procedure dbnvgr51Click(Sender: TObject; Button: TNavigateBtn);
procedure dbgrd55MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Pulizia (Sender: TObject);
procedure dbgrd51CellClick(Column: TColumn);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Edit3Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form5: TForm5;
    AnaGAFlg:Boolean=True;
    inserisco5:Boolean=False;
    CanClose:Boolean=True;
    filtro,sg,tabl,garastr,giornostr:string;
    tabella:TDataset;

implementation
uses
    Console1;

    {$R *.dfm}
    {$D+}
procedure TForm5.FormCreate(Sender: TObject);    //Crea il Form5
begin
    //Form2.ABSTable5.TableName:='AnaGa';
    Form2.ABSTable5.Open;
    Form2.ABSTable5.Edit;
    // Form2.ABSTable6.TableName:='Base';
    Form2.ABSTable6.Open;
    Form2.ABSTable6.Edit;
    AnaGAFlg:=False;
    ds51.DataSet:=Form2.ABSTable5;//aggiunto per evitare circular DataLink error
    ds52.DataSet:=Form2.ABSTable4;
    // Form2.ABSTable4.TableName:='AnaDspl';
    Form2.ABSTable4.Open;
    Form2.ABSTable4.Edit;
    //Form2.ABSTable8.TableName:='GRPA';
    Form2.ABSTable8.Open;
    Form2.ABSTable8.Edit;
    //Form2.ABSTable9.TableName:='GRPB';
    Form2.ABSTable9.Open;
    Form2.ABSTable9.Edit;
    shp1.Brush.Color:=clOlive;    //per colorare la shape

```

```

    shp2.Brush.Color:=clRed; //per colorare la shape
    Form2.ABSTable1.Open; //registra su Servizio
    Form2.ABSTable1.INSERT;
    Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
    Form2.ABSTable1.FieldName('what').AsString:='Accesso all''Anagrafica Gare';
    Form2.ABSTable1.FieldName('OUT').AsString:='';
    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
    TDateTimeField(Form2.ABSTable4.FieldName('TMAX')).
        DisplayFormat:='nn:ss,zzz';
end;

```

```

procedure TForm5.Button1Click(Sender: TObject); //Esci
begin
    if inserisco5=True then //se ha inserito un record
        begin
            FormCloseQuery(Self,canclose);
        end
    else
        begin
            Form5.Pulizia(Self); //occorre verificare se i db sono già chiusi
            Form2.ABSTable1.Open;
            Form2.ABSTable1.Last;
            if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
                Form2.ABSTable1.Insert;
            Form2.ABSTable1.Edit;
            Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
            Form2.ABSTable1.Post;
            Form2.ABSTable1.Close;
            Form2.ABSTable4.Close;
            Form2.ABSTable5.Close;
            Form5.Close; // NB chiama il can close
            AnaGAFlg:=True; //per consentire di monitorizzare che Form5 è closed
            Form2.ABSTable6.Close;
            Form2.ABSTable8.Close;
            Form2.ABSTable9.Close;
            Form2.ABSTable10.Close;
            Form2.ABSTable10.EmptyTable;
            Form2.ABSTable11.Close;
            Form2.ABSTable11.EmptyTable;
            Form2.ABSTable12.Close;
            Form2.ABSTable12.EmptyTable;
            Form2.Visible:=True;
        end;
    end;

```

```

procedure TForm5.Button2Click(Sender: TObject); //CERCA
label
vai;
var
chi:array[1..2]of shortint;
i:ShortInt;
begin
    ds51.DataSet:=ABSQuery6;
    i:=0; //determina se data e/o luogo è in ricerca
    if edit1.text<>' ' then

```

```

begin
    i:=i+1;
    chi[i]:=1;
end;
if edit2.Text<>' ' then
begin
    i:=i+1;
    chi[i]:=2;
end;
    if chi[2]>0 then //elabora i query
begin
    filtro:='DATA LIKE ' + QuotedStr('%'+Edit1.Text+'%')+ ' AND LUOGO LIKE '
    +quotedStr('%'+edit2.text+'%');
    goto vai;
end;
    if chi[1]>0 then
begin
    if (edit1.Text<>' ') then
        filtro:= 'DATA LIKE '+quotedstr('%'+Edit1.Text+'%');
    if (edit2.Text<>' ') then
        filtro:='LUOGO LIKE '+QuotedStr('%'+Edit2.Text+'%');
    goto vai;
end;
vai :
    //effettua il query
    with ABSQuery6 do
begin
    close;
    if i>0 then
begin
        SQL.Text := 'select * from AnaGA where '+filtro; //Attenzione agli spazi
prima e dopo gli apici
        ExecSql;
    end else
begin
        ShowMessage('Nessun elemento di ricerca inserito');
        Exit;
    end;
    Open;
end;
end;

procedure TForm5.Button3Click(Sender: TObject); //Annulla
begin
    Edit1.Text:='';
    Edit2.Text:='';
    ds51.DataSet:=Form2.ABSTable5;
end;

procedure TForm5.Button4Click(Sender: TObject); //AZZERAZIONE
begin
    if MessageDlg('Stai per azzerare l''intera Anagrafe delle Gare. Vuoi continuare
?',
        mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
begin
    Form2.ABSTable5.Close;
    Form2.ABSTable5.EmptyTable;
end;
end;

```

```

        ShowMessage('Tutta l'anagrafe è stata cancellata !');
    end;
end;

procedure TForm5.cbb55Change(Sender: TObject); // Seleziona un dato della Combobox
begin
    //effettuando una scelta viene definita la Table attiva
    if cbb55.Text='Base' then
    begin
        tabella:=Form2.ABSTable6;
        tabl:='Base';
        ds55.DataSet:=tabella;
        ShowMessage('Crea la Tabella Discipline Base'+#13+
        'Clicca sul LUOGO della Gara'+#13+'per inserire il suo codice.');// Tab
        end;
    if cbb55.Text='GruppoA' then
    begin
        tabella:=Form2.ABSTable8;
        tabl:='GRPA';
        ds55.DataSet := tabella;
        ShowMessage('Crea la Tabella Discipline Gruppo A '); //Tab 8
        end;
    if cbb55.Text='GruppoB' then
    begin
        tabella:=Form2.ABSTable9;
        tabl:='GRPB';
        ds55.DataSet := tabella; //in questo modo mostra gli accoppiamenti già
        esistenti
        ShowMessage('Crea la Tabella Discipline Gruppo B '); //Tab 9
        end;
    end;

    //Accoppiamento GARA - DISCIPLIN
    // Inserimento codice Gara
procedure TForm5.dbgrd51CellClick(Column: TColumn);
Var
    CoGaSi:Boolean;
begin
    Self.ActiveControl := nil; //per togliere eventuali focus precedenti
    if cbb55.ItemIndex>-1 then
    begin
        sg:= Form2.ABSTable5.FieldName('CODGA').AsString;
        //verifica l'esistenza di un precedente accoppiamento per quella Gara
        ds55.DataSet:=ABSQuery6; //nella tabella degli accoppiamenti (TA)
        with ABSQuery6 do
        begin
            Close; //inserire una variabile nel query
            SQL.Text := 'Select * from '+tabl+' where CODGA = '+ QuotedStr(sg);
            ExecSql;
            Open;
            end;
        if ABSQuery6.RecordCount>0 then //nel caso ci sia già la GARA nella tabella
            if MessageDlg('Hai già elaborato questa Gara '+#13+'Vuoi aggiungere
            discipline ?',
            mtconfirmation, [mbYes,mbNo], 0) = mrNo then
            begin
                if MessageDlg('Vuoi Rifarla del tutto ?',mtconfirmation,

```

```

[mbYes,mbNo], 0) = mrYes then
    begin
        Form2.ABSTable6.Edit;
        with Form2.ABSTable6 do // nel Query c'è già l'elenco degli
accoppiamenti di quella Gara
        begin
            First;
            while(not Eof)do
            begin
                //seleziona e cancella
                CoGaSi:= Form2.ABSTable6.Locate('CODGA',sg,[]);
                if CoGaSi=True then Delete;
                Next;
            end;
        end;
        end;
        ds55.DataSet:= tabella;
        Exit;
    end; //showmessage con posizione
    ShowMessagePos('Questi sono gli accoppiamenti già inseriti',650,250);

    ds55.DataSet:= tabella; //ritorna al dataset primario
    if MessageDlg('Codice Gara = '+sg+' ?', mtconfirmation, [mbYes,mbNo], 0) =
mrYes then
    begin
        tabella.Insert;
        tabella.FieldName('CODGA').AsInteger:=strToInt(sg);
        tabella.Post;
        showmessage('Clicca sulla DESCRIZIONE della disciplina da accoppiare');
    end else
    begin
        sg:='';
        Exit ;
    end;
end
else
    if inserisco5=False then
        ShowMessage('Effettua prima una scelta di Gruppo');
    end;

procedure TForm5.dbgrd52MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer); //Accoppiamento GARA - DISCIPLINE
var
    s:string; //Inserimento codice DISCIPLINE
begin
    if cbb55.ItemIndex>-1 then
        begin
            if sg='' then
                begin
                    //per definire su quale gara inserire le DSPL
                    showmessage('Clicca prima sul luogo della GARA');
                    Exit;
                end;
            s:= Form2.ABSTable4.FieldName('CODDSPL').AsString;
            if MessageDlg('Codice Disciplina = '+s+' ?' , mtconfirmation,
[mbYes,mbNo], 0) = mrYes then
                begin
                    //verifica che non sia già inserito
                    ds55.DataSet:=ABSQuery6;
                    with ABSQuery6 do

```

```

begin
    Close; //inserire una variabile nel query
    SQL.Text := 'Select * from '+tabl+' where CODGA = '+
QuotedStr(sg)+
    ' and CODDSPL = '+ QuotedStr(s);
    ExecSql;
    Open;
end;
if ABSQuery6.RecordCount>0 then //caso in cui c'è già la DSPL in
TA
    begin
        ShowMessage('ERRORE - Questa disciplina è già inserita');
        ds55.DataSet:=tabella; //riporta il dbgrid a leggere i
        valori
        s:='';
        Exit;
    end;
ds55.DataSet:=tabella;
tabella.Edit;
tabella.FieldName('CODDSPL').AsInteger:=strToInt(s);
tabella.Post;
s:='';
if MessageDlg('Vuoi inserire altre Discipline'+#13+'nella stessa
Gara ?' ,
    mtconfirmation, [mbYes,mbNo], 0) = mrYes then
    begin
        tabella.Insert;
        tabella.FieldName('CODGA').AsInteger:=strToInt(sg);
        tabella.Post;
        showmessage('Clicca sulla DESCRIZIONE della disciplina da
accoppiare');
    end else
    begin
        sg:=''; //azzera il codice gara precedente
        s:='';
        ds55.DataSet:=tabella;
        Exit;
    end;
end
else Exit;

end
else ShowMessage('Effettua prima una scelta di tabella');
end;

procedure TForm5.dbgrd55MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer); //verifica la scelta
begin
    if cbb55.ItemIndex=-1 then
        ShowMessage('Effettua prima una scelta di tabella');
    end;

procedure TForm5.dbnvgr51Click(Sender: TObject; Button: TNavigateBtn);
begin //intercetta il dbnavigator
    case button of
        nbInsert: //se inserisce un nuovo record
        begin
            Form2.ABSTable5.Last;

```

```

Form2.ABSTable5.Append;
inserisco5:=True; //inserisce il flag nel nuovo record
dbgrd51.SetFocus;
DBGrd51.SelectedField := Form2.ABSTable5.FieldName('DATA');
end ;
nbPost:
begin //se lo carica nella tabella
    inserisco5:=False ;
end;
nbDelete: //se cancella il record - serve anche per evitare di entrare in
Query
begin
    inserisco5:=False;
end;
end;
end;

```

```

procedure TForm5.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin //Chiusura del Form5 Gara
    if inserisco5=True then
        if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) =
mrYes
        then
            begin
                CanClose:=False;
                Exit;
            end
        else
            begin
                CanClose:=True;
                if inserisco5=True then //conferma il Falso di inserisco5
                    inserisco5:=False;
                    Button1Click(Self);
                end;
            end;
end;

```

```

procedure TForm5.Edit3Click(Sender: TObject); //Acquisisce la gara per il report
begin // ripristina le tabelle
    Edit3.Text:='';
    Form2.ABSTable10.Close;
    Form2.ABSTable10.EmptyTable;
    Form2.ABSTable11.Close;
    Form2.ABSTable11.EmptyTable;
    Form2.ABSTable12.Close;
    Form2.ABSTable12.EmptyTable;
    // Form2.ABSTable10.TableName:='repBase';
    Form2.ABSTable10.Open;
    Form2.ABSTable10.Edit;
    // Form2.ABSTable11.TableName:='repGRPA';
    Form2.ABSTable11.Open;
    Form2.ABSTable11.Edit;
    // Form2.ABSTable12.TableName:='repGRPB';
    Form2.ABSTable12.Open;
    Form2.ABSTable12.Edit;

```

```

end;
procedure TForm5.Button5Click(Sender: TObject);    //prepara il report
label
GA,GB,FINE;
var
codgasi,dsplsi:Boolean;
codgan,DSPLn:Integer;
DSPLstr:string;
begin    //Riempire le 3 tabelle
    if Edit3.Text='' then
        begin
            ShowMessage('Devi inserire il codice della Gara');
            Exit;
        end else    //il codice GARA è presente
        begin    // Verifica l'esistenza della Gara negli accoppiamenti Base etc.

            codgasi:= Form2.ABSTable5.Locate('CODGA',Edit3.text,[]);
            if codgasi=False then
                begin
                    ShowMessage('Questo codice Gara è inesistente ');
                    Exit;
                end else
                begin    //Trovata la gara ed il giorno di esecuzione
                    codgan:=StrToInt(Edit3.Text);
                    garastr:= Form2.ABSTable5.FieldName('LUOGO').AsString;
                    giornostr := Form2.ABSTable5.FieldName('DATA').AsString;
                end;

                //trovare le discipline della BASE
            if Form2.ABSTable10.RecordCount>0 then    //la tabella Base è già piena
                begin
                    if MessageDlg('Il report delle discipline Base è già fatto. Vuoi
rifarlo?' ,
                        mtconfirmation, [mbYes,mbNo], 0) = mrYes then
                        begin
                            Form2.ABSTable10.EmptyTable;
                        end else goto GA;
                    end;
                    Form2.ABSTable6.DisableControls;
                    try
                        Form2.ABSTable6.First;
                        while not Form2.ABSTable6.Eof do
                            begin    //per ogni codga ricavare il cod dspl e memorizzare la DESCR
nella Tabella
                                if codgan=Form2.ABSTable6.FieldName('CODGA').AsInteger then
                                    begin    //se il codice gara corrisponde

                                        DSPLn:=Form2.ABSTable6.FieldName('CODDSPL').AsInteger;//memorizza la
                                        disciplina
                                        dsplsi:=Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]);//cerca
                                        la disciplina sull'anagrafica
                                        if dsplsi=True then //quando la trova
                                            begin
                                                DSPLstr:=
Form2.ABSTable4.FieldName('DESCR').AsString;//memorizza la descrizione
                                                Form2.ABSTable10.Insert;    //la registra sulla tabella
del report

```



```

Form2.ABSTable10.FieldByName('CODDSPL').AsInteger:=DSPLn;

Form2.ABSTable10.FieldByName('DESCR').AsString:=DSPLstr;

Form2.ABSTable10.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('
TMAX').AsDateTime;

        Form2.ABSTable10.Post;
        end;
    end;
    Form2.ABSTable6.Next;
    end;
finally
    Form2.ABSTable6.EnableControls;
    ShowMessage('Registrate le discipline Base');
end;

GA :
        //trovare le discipline del GruppoA
if Form2.ABSTable11.RecordCount>0 then //èvuota la tabella GRPA ?
    begin
        if MessageDlg('Il report delle discipline Gruppo A è già fatto. Vuoi
rifarlarlo?' ,
            mtconfirmation, [mbYes,mbNo], 0) = mrYes then
            begin
                Form2.ABSTable11.EmptyTable;
            end else goto GB;
        end;

Form2.ABSTable8.DisableControls;
try
    Form2.ABSTable8.First;
    while not Form2.ABSTable8.Eof do
        begin //per ogni codga ricavare il cod dspl e memorizzare la DESCR
nella Tabella
            if codgan=Form2.ABSTable8.FieldByName('CODGA').AsInteger then
                begin //se il codice gara corrisponde

DSPLn:=Form2.ABSTable8.FieldByName('CODDSPL').AsInteger;//memorizza la
disciplina
                dsplsi:=Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]);//cerca
la disciplina sull'anagrafica
                if dsplsi=True then //quando la trova
                    begin
                        DSPLstr:=
Form2.ABSTable4.FieldByName('DESCR').AsString;//memorizza la descrizione
                        Form2.ABSTable11.Insert; //la registra sulla tabella
del report

Form2.ABSTable11.FieldByName('CODDSPL').AsInteger:=DSPLn;

Form2.ABSTable11.FieldByName('DESCR').AsString:=DSPLstr;

Form2.ABSTable11.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('
TMAX').AsDateTime;

                        Form2.ABSTable11.Post;
                        end;
                    end;
                Form2.ABSTable8.Next;

```

```

        end;
    finally
        Form2.ABSTable8.EnableControls;
        ShowMessage('Registrate le discipline Gruppo A');
    end;

    GB :

    if Form2.ABSTable12.RecordCount>0 then //èvuota la tabella Base ?
    begin
        if MessageDlg('Il report delle discipline Gruppo B è già fatto. Vuoi rifarlo?' ,
            mtconfirmation, [mbYes,mbNo], 0) = mrYes then
            begin
                Form2.ABSTable12.EmptyTable;
            end else goto FINE;
        end;

        Form2.ABSTable9.DisableControls;
        try
            Form2.ABSTable9.First;
            while not Form2.ABSTable9.Eof do
                begin //per ogni codga ricavare il cod dspl e memorizzare la DESCR
                    nella Tabella
                        if codgan=Form2.ABSTable9.FieldName('CODGA').AsInteger then
                            begin //se il codice gara corrisponde
                                DSPLn:=Form2.ABSTable9.FieldName('CODDSPL').AsInteger;//memorizza la
                                disciplina
                                dsplsi:=Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]);//cerca
                                la disciplina sull'anagrafica
                                if dsplsi=True then //quando la trova
                                    begin
                                        DSPLstr:=
                                        Form2.ABSTable4.FieldName('DESCR').AsString;//memorizza la descrizione
                                        Form2.ABSTable12.Insert; //la registra sulla tabella
                                        del report

                                        Form2.ABSTable12.FieldName('CODDSPL').AsInteger:=DSPLn;

                                        Form2.ABSTable12.FieldName('DESCR').AsString:=DSPLstr;

                                        Form2.ABSTable12.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName('
                                        TMAX').AsDateTime;

                                        Form2.ABSTable12.Post;
                                    end;
                                end;
                                Form2.ABSTable9.Next;
                            end;
                        finally
                            Form2.ABSTable9.EnableControls;
                            ShowMessage('Registrate le discipline Gruppo B');
                        end;

                        FINE :
                    end;
                // Presentare il report Nota bene come si passano le variabili
                Form5.frxReport5.Variables['mia1'] := QuotedStr(giornostr);

```

```

        Form5.frxReport5.Variables['mia2'] := QuotedStr(garastr);
        // Form5.frxPDFExport51.FileName:='Accoppiamenti';
        Form5.frxReport5.ShowReport;
        Edit3.Text:='';
    end;

    procedure TForm5.Button6Click(Sender: TObject);
    begin
        Form5.frxReport5.Print;
    end;

    // ELENCA o STAMPA L'anagrafe Gare
    procedure TForm5.Button7Click(Sender: TObject);
    begin
        Form5.frxReport51.ShowReport;
        if MessageDlg('Vuoi stamparlo ?' , mtconfirmation, [mbYes,mbNo], 0,mbNo) =
mrNo
            then Exit else
                Form5.frxReport51.Print;
    end;

    procedure TForm5.Button8Click(Sender: TObject); //Azzerà gruppo Base
    begin
        if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo Base. Vuoi
continuare ?',
            mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
            begin
                Form2.ABSTable6.Close;
                Form2.ABSTable6.EmptyTable;
                ShowMessage('Tutta l''anagrafe è stata cancellata !');
            end;
    end;

    procedure TForm5.Button9Click(Sender: TObject); // Azzerà gruppo A
    begin
        if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo A. Vuoi
continuare ?',
            mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
            begin
                Form2.ABSTable8.Close;
                Form2.ABSTable8.EmptyTable;
                ShowMessage('Tutta l''anagrafe è stata cancellata !');
            end;
    end;

    procedure TForm5.Button10Click(Sender: TObject); //Azzerà gruppo B
    begin
        if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo B. Vuoi continuare
?',
            mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
            begin
                Form2.ABSTable9.Close;
                Form2.ABSTable9.EmptyTable;
                ShowMessage('Tutta l''anagrafe è stata cancellata !');
            end;
    end;

    //Pulizia dei record vuoti

```

```

procedure TForm5.Pulizia(Sender:TObject);  //Quando crei una Procedura premetti
il Form
var
trovato,rimosso:Boolean ;
begin
    trovato:=False;
    rimosso:=False;
    if Form2.ABSTable6.RecordCount>0 then
begin
        trovato:=Form2.ABSTable6.Locate('CODDSPL',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable6.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        trovato:=Form2.ABSTable6.Locate('CODGA',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable6.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        end;
end;

    if Form2.ABSTable8.RecordCount>0 then
begin
        trovato:=Form2.ABSTable8.Locate('CODDSPL',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable8.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        trovato:=Form2.ABSTable8.Locate('CODGA',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable8.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        end;
end;

    if Form2.ABSTable6.RecordCount>0 then
begin
        trovato:=Form2.ABSTable9.Locate('CODDSPL',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable9.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        trovato:=Form2.ABSTable9.Locate('CODGA',null,[]);  //senza LocateOptions
        if trovato=True then
            begin
                Form2.ABSTable9.Delete;
                rimosso:=True;
                trovato:=False;
            end;
        end;
end;

```

```
    end;  
end;  
  
if rimosso=True then  
    begin  
        ShowMessage( 'Rimosso ogni record senza codice Gara o Disciplina' );  
        rimosso:=False;  
    end;  
end;  
end.  
end.
```

unit Gara;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Mask, Vcl.DBCtrls,
Vcl.ExtCtrls, Vcl.Grids, Vcl.DBGrids, Data.DB, ABSSMain, Vcl.ComCtrls,
Vcl.Imaging.jpeg, Vcl.Buttons, frxClass, frxDBSet, frxExportPDF,
Datanap.Provider, Datanap.DBClient, System.UITypes;

type

TForm6 = **class** (TForm)
 Button1: TButton;
 Button2: TButton;
 Button3: TButton;
 Button4: TButton;
 dbgrd72: TDBGrid;
 dbnvgr7: TDBNavigator;
 ds7: TDataSource;
 ds72: TDataSource;
 AbsQuery72: TABSQuery;
 cbb72: TComboBox;
 Label7: TLabel;
 Timer1: TTimer;
 Label8: TLabel;
 shp1: TShape;
 Label9: TLabel;
 dbgrd7: TDBGrid;
 dbgrd74: TDBGrid;
 ds74: TDataSource;
 Label11: TLabel;
 Edit1: TEdit;
 Edit2: TEdit;
 ABSQuery7: TABSQuery;
 Button5: TButton;
 ABSQuery74: TABSQuery;
 img2: TImage;
 cbb74: TComboBox;
 Label2: TLabel;
 rb1: TRadioButton;
 rb2: TRadioButton;
 btn1: TBitBtn;
 img3: TImage;
 Label3: TLabel;
 Label4: TLabel;
 frxdbdtst13: TfrxDBDataset;
 ABSQuery73: TABSQuery;
 ABSTable13: TABSTable;
 ds13: TDataSource;
 frxrprt7: TfrxReport;
 frxPDFExport7: TfrxPDFExport;
 Button6: TButton;
 Edit3: TEdit;
 Button7: TButton;
 ABSTable14: TABSTable;
 Button8: TButton;

```

Bevel1: TBevel;
Label6: TLabel;
Button9: TButton;
Edit4: TEdit;
Label10: TLabel;
Button10: TButton;
frxReport14: TfrxReport;
frxDBDataset14: TfrxDBDataset;
ABSQueryDUP: TABSQuery;
ABSQueryDUP2: TABSQuery;
Bevel2: TBevel;
Label5: TLabel;
CheckBox1: TCheckBox;
Button11: TButton;
CheckBox2: TCheckBox;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure dbgrd72CellClick(Column: TColumn);
procedure Button2Click(Sender: TObject);
procedure dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);
procedure cbb72Select(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure Button5Click(Sender: TObject);
procedure cbb74Change(Sender: TObject);
procedure rb2Click(Sender: TObject);
procedure rb1Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure btn1Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure ABSTable14BeforeDelete(DataSet: TDataSet);
procedure Button9Click(Sender: TObject);
procedure dbgrd7CellClick(Column: TColumn);
// procedure GetText(Sender: TField; var Text: string; DisplayText: Boolean);
procedure Button10Click(Sender: TObject);
procedure dbnvgr7BeforeAction(Sender: TObject; Button: TNavigateBtn);
procedure CheckBox1Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
//NOTA BENE l'entry del codice gara non triggera alcun evento
var
    Form6: TForm6;
    GaFlg:Boolean =True;
    inserisco7:Boolean=False;
    ATL:Integer;
    scelta:Boolean=False; // se si sceglie con la combobox > True
    filtGa:Boolean=False; // idem se filtra Gara
    CanClose:Boolean=True;
    LastGa:SmallInt;

```

```

numgara:Integer;           //T7 dati gare mai cancellata
tempo:string;
mancante:Integer;
manleva:boolean;          //T4 Anagrafica discipline cancellata in
AnaDspl
    rab1,rab2,rab3,mncat,mnpos,mnpet,mnmanl,mncti:Boolean; //T5 Anagrafica Gare
cancellata in AnaGa
implementation             //T13 dati per report - temporanea
    uses                   //T14 dati buffer - temporanea
        Console1,aiuto,Tempi , Mancanti;
{$R *.dfm}
{$D+}
procedure TForm6.FormCreate(Sender: TObject); //Crea il form
begin
    // Form2.ABSTable7.TableName:='Gara';
    Form2.ABSTable7.Open;                                     //FORMATTARE CAMPI DBGRID

    TDateTimeField(Form2.ABSTable7.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';

    TDateTimeField(Form2.ABSTable7.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';

    TDateTimeField(Form2.ABSTable7.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
    TBooleanField(Form2.ABSTable7.FieldByName('DQ')).DisplayValues:='Si;No';
    TBooleanField(Form2.ABSTable7.FieldByName('MANL')).DisplayValues:='Si;No';
    ABSTable14.TableName:='Corrente';
    ABSTable14.Open;
    ABSTable13.TableName:='Maestra';
    TDateTimeField(ABSTable14.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
    TDateTimeField(ABSTable14.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
    TDateTimeField(ABSTable14.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
    GaFlg:=False;
    //ds7.DataSet:=Form2.ABSTable7; //aggiunto per evitare circular DataLink error
    Form2.ABSTable7.Edit;
    Form2.ABSTable4.Open;
    Form2.ABSTable1.Open;
    Form2.ABSTable1.INSERT;
    Form2.ABSTable1.FieldByName('IN').AsDateTime:= NOW;
    Form2.ABSTable1.FieldByName('what').AsString:='Accesso alle Iscrizioni';
    Form2.ABSTable1.FieldByName('OUT').AsString:='';
    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
    cbb74.Text:='Scegli';
    Edit1.Text:='';
    rb1.Checked:=True ;
    Edit2.Visible:=False;
    Label1.Visible:=False;
    Form2.ABSTable5.TableName:='AnaGa';
    Form2.ABSTable5.Open;
    Form2.ABSTable5.Last;
    LastGa:=Form2.ABSTable5.FieldByName('CODGA').AsInteger ;
    Label4.Caption:=IntToStr(LastGa);
    Button2.Click; //Refresh
    Button5.Click; //in tal modo all'apertura viene evidenziato lo storico
    ATL:=0;
    manleva:=False; //inizializzare
end;

```



```

    {procedure TForm6.GetText(Sender: TField; var Text: string; DisplayText:
Boolean);
begin    //dovrebbe servire per scrivere yes nel campo DQ, quando c'è
    if ABSTable14.FieldName('DQ').AsBoolean then
        Text := 'Si'
    else
        Text := 'No';

end;
}

procedure TForm6.rb1Click(Sender: TObject); //AUTOMATICA
begin
    Edit2.Visible:=False;    //toglie edit2
    Label1.Visible:=False;
    cbb74.Visible:=True;
    Label2.Visible:=True;
end;

procedure TForm6.rb2Click(Sender: TObject); // MANUALE
begin
    if Edit1.Text='' then
    begin
        ShowMessage('Inserisci il Codice della Gara ');
        rb1.Checked:=True;    //rb1 automatico
        Exit;
    end;
    cbb74.Visible:=False;
    Label2.Visible:=False;
    Edit2.Visible:=True;
    Label1.Visible:=True;
    ds74.DataSet:=Form2.ABSTable4;
end;

procedure TForm6.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    if inserisco7=True then
    if MessageDlg('Hai inserito un dato-gara. Sincronizzi ?' , mtconfirmation,
[mbYes,mbNo], 0) = mrYes
    then
    begin
        CanClose:=False;
        Exit;
    end
    else
    begin
        CanClose:=True;
        if inserisco7=True then //conferma il Falso di inserisco7
            inserisco7:=False;
            Button1Click(Self);
        end;
    end;
end;

procedure TForm6.Timer1Timer(Sender: TObject);    //Flashing
begin
    if Edit1.Text='' then

```

```

begin
    if label8.Font.Color = clRed then
        label8.Font.Color := clNavy
    else
        label8.Font.color := clRed;
    end else Label8.Font.Color:=clGray;

if Edit2.Text='' then
begin
    if label11.Font.Color = clRed then
        label11.Font.Color := clNavy
    else
        label11.Font.color := clRed;
    end else Label11.Font.Color:=clGray;

if Edit4.Text='' then
begin
    if label10.Font.Color = clRed then
        label10.Font.Color := clNavy
    else
        label10.Font.color := clRed;
    end else Label10.Font.Color:=clGray;

    //condizioni per garantire che label2 sia invisibile, altrimenti il colore
vince
if ((cbb74.ItemIndex=0) or (cbb74.Text='Scegli'))and (cbb74.Visible=True) then
begin
    if label2.Font.Color = clRed then
        label2.Font.Color := clNavy
    else
        label2.Font.color := clRed;
        Label2.Visible:=True;
    end else Label2.Font.Color:=clGray;

end;

procedure TForm6.ABSTable14BeforeDelete(DataSet: TDataSet);
begin
    if ds7.DataSet=ABSTable14 then
        begin
            ShowMessage('Attenzione-Se vuoi rimuovere un record, devi farlo sullo Storico
!' + #13 +
            'Clicca Storico ed identifica il record con il suo codice');
            Abort;
            Exit;
        end;
end;

procedure TForm6.btn1Click(Sender: TObject);
begin
    // Form7.Create(self);
    Form7.Show;
end;

procedure TForm6.Button1Click(Sender: TObject);    //Esci

begin

```

```

    if filtGa =True then
    if MessageDlg('Devi sincronizzare la tabella delle iscrizioni ?'
, mtconfirmation, [mbYes,mbNo], 0) = mrYes then Exit;

if inserisco7=True then //se ha inserito un record
begin
    FormCloseQuery(Self,canclose);
end
else
begin
    filtGa:=False;
    Form2.ABSTable7.Close;
    Form6.Close; // NB chiama il can close
    Form9.Close;
    GaFlg:=True; //per consentire di monitorizzare che Form6 èclosed
    Form2.Visible:=True;
    Edit1.Text:=''; //rimuove il codice della gara
    Edit2.Text:=''; // '' della DSPL
    Form2.ABSTable10.Close; //Chiude non cancella
    Form2.ABSTable11.Close;
    Form2.ABSTable12.Close;
    Form6.ABSTable13.Close;
    Form6.ABSTable13.EmptyTable;
    Form6.ABSTable14.Close;
    Form6.ABSTable14.EmptyTable;
    Form2.ABSTable2.Close;
    Form2.ABSTable4.Close;
    Form2.ABSTable5.Close;
    Form2.ABSTable6.Close;
    Form2.ABSTable8.Close;
    Form2.ABSTable9.Close;
    Form10.Close;
    Form2.ABSTable1.Open;
    Form2.ABSTable1.Last;
    if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
        Form2.ABSTable1.Insert;
    Form2.ABSTable1.Edit;
    Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
end;

end;

procedure TForm6.Button2Click(Sender: TObject); //Refresh dbgrid Atleti
begin
    ds72.DataSet:=AbsQuery72;
    with AbsQuery72 do
    begin
        Close; //inserisce una variabile nel query
        SQL.Text := 'Select * from AnaAtl';
        ExecSql;
        Open;
    end;
end;

procedure TForm6.Button3Click(Sender: TObject); //Rapporto

```

```

var
CODGAn:Integer;
DSPLsi,ATLsi:Boolean;
CODDSPL13,CODATL13:Integer;
begin
    Form2.ABSTable4.Open;
    //Form2.ABSTable4.Edit;
    Form2.ABSTable2.Open;
    //Form2.ABSTable2.Edit;
    CODGAn:=StrToInt(inputBox('Rapporto Iscrizioni','Inserire il codice della
Gara',Edit1.Text));
    ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
    with ABSQuery7 do
        begin
            Close; //porto su DBGRID solo la gara scelta con tutti gli
iscritti
            SQL.Text := 'Select * from Gara where CODGA =
'+QuotedStr(IntToStr(CODGAn))+ ' ORDER BY CODATL Asc, CODDSPL Asc';
            ExecSql;
            Open; //Estrae i dati di iscrizione solo per una gara
            First;
            // La ricostruisce da cleaned
            Form6.ABSTable13.Close;
            Form6.ABSTable13.EmptyTable;
            Form6.ABSTable13.Open;

TBooleanField(ABSTable13.FieldByName('DQ')).DisplayValues:='Si;No';

TDateTimeField(ABSTable13.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';

TDateTimeField(ABSTable13.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';

TDateTimeField(ABSTable13.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';

TBooleanField(Form6.ABSTable13.FieldByName('MANL')).DisplayValues:='Si;No';
    Form6.ABSTable13.Edit; // Crea Tabella Master per
il report = Maestra
    while (not Eof) do //sempre sotto ABSQuery7
    begin // copia il Query in una
Table
        Form6.ABSTable13.Insert;

Form6.ABSTable13.FieldByName('CODGA').AsInteger:=ABSQuery7.FieldByName('CODGA'
).AsInteger;

Form6.ABSTable13.FieldByName('CODDSPL').AsInteger:=ABSQuery7.FieldByName('CODD
SPL').AsInteger;

Form6.ABSTable13.FieldByName('CODATL').AsInteger:=ABSQuery7.FieldByName('CODAT
L').AsInteger;

Form6.ABSTable13.FieldByName('COGNOME').AsString:=ABSQuery7.FieldByName('COGNO
ME').AsString;

Form6.ABSTable13.FieldByName('NMPET').AsInteger:=ABSQuery7.FieldByName('NMPET'
).AsInteger;

Form6.ABSTable13.FieldByName('NMPOS').AsInteger:=ABSQuery7.FieldByName('NMPOS'

```

```

).AsInteger;

Form6.ABSTable13.FieldByName('TAME1').AsDateTime:=ABSQuery7.FieldByName('TAME1')
.AsDateTime;

Form6.ABSTable13.FieldByName('TBME2').AsDateTime:=ABSQuery7.FieldByName('TBME2')
.AsDateTime;

Form6.ABSTable13.FieldByName('DQ').AsBoolean:=ABSQuery7.FieldByName('DQ').AsBo
olean;

Form6.ABSTable13.FieldByName('PEN').AsDateTime:=ABSQuery7.FieldByName('PEN').A
sDateTime;

Form6.ABSTable13.FieldByName('MANL').AsBoolean:=ABSQuery7.FieldByName('MANL').
AsBoolean;

Form6.ABSTable13.FieldByName('CAT').AsInteger:=ABSQuery7.FieldByName('CAT').As
Integer;

        Form6.ABSTable13.Post;
        ABSQuery7.Next;
    end;

    end;          //aggiungo i campi DESCR e TMAX
ABSTable13.First;
while not ABSTable13.EOF do
begin
    CODDSPL13:=ABSTable13.FieldByName('CODDSPL').AsInteger;
    DSPLsi:=Form2.ABSTable4.Locate('CODDSPL',CODDSPL13,[] );//si blocca
sul record con pari codice DSPL
    if DSPLsi=False then ShowMessage('Valore non trovato'); //NB nel locate
il valore da ricercare èvariant (non stringa)
    ABSTable13.Edit;

ABSTable13.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName('DESCR')
.AsString;

ABSTable13.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('TMAX')
.AsDateTime;
    ABSTable13.Post;
    ABSTable13.NEXT
end;
// Aggiungo il nome e nazione e il telefono
ABSTable13.First;
while not ABSTable13.EOF do
begin
    CODATL13:=ABSTable13.FieldByName('CODATL').AsInteger;
    ATLsi:=Form2.ABSTable2.Locate('CODATL',CODATL13,[] );//si blocca sul
record con pari codice DSPL
    if ATLsi=False then ShowMessage('Valore non trovato');
// NB nel locate il valore da ricercare èvariant (non stringa)
    ABSTable13.Edit;

ABSTable13.FieldByName('NOME').AsString:=Form2.ABSTable2.FieldByName('NOME').A
sString;

ABSTable13.FieldByName('NAZIONE').AsString:=Form2.ABSTable2.FieldByName('NAZIO
NE').AsString;

```

```

ABSTable13.FieldName('TLF').AsString:=Form2.ABSTable2.FieldName('TLF').AsString;
    ABSTable13.Post;
    ABSTable13.NEXT
end;
Form6.frxrprt7.ShowReport;
end;

procedure TForm6.Button4Click(Sender: TObject); //Stampa Report
begin
    Form6.frxrprt7.Print;
end;

procedure TForm6.Button5Click(Sender: TObject); //Refresh tabella iscrizioni -
STORICO
begin
    DBNvgr7.VisibleButtons:=[nbFirst,nbPrior,nbNext,nbLast,nbDelete];
    dbgrd7.Columns[0].FieldName:='NMREC'; //applicare il campo relativo al
cambio di Dataset
    Label6.Caption:='Dati di tutte le Gare presenti in memoria';
    ds7.DataSet:=Form2.ABSTable7;
    DBgrd7.ReadOnly:=True;
    FiltGa:=False;
    if Form9.Visible=True then
        Form9.Hide;
        checkbox2.Checked:=False;
        Form10.Hide;
        ABSTable14.Close;
        ABSTable14.EmptyTable;
    end;

procedure TForm6.Button6Click(Sender: TObject); // Filtro una gara
LAVORI IN CORSO CODICE DA RIVEDERE
var
    quanti,btscelta :Integer;
    cosa:TStringList;
    ca,cd,ga:Integer;
begin
    DBNvgr7.VisibleButtons:=[nbFirst,nbPrior,nbNext,nbLast];
    if Edit3.Text='' then
        begin
            ShowMessage('Inserisci il codice della Gara nella casella a destra !');
            Exit;
        end;
    Form6.ABSTable14.TableName:=('Corrente');
    Form6.ABSTable14.Close;
    Form6.ABSTable14.EmptyTable;
    Form6.ABSTable14.Open;
    Form6.ABSTable14.Edit;
    quanti:=ABSTable14.RecordCount;
    filtGa:=True;
    if (quanti>0) AND (mncti=False) then //Continua solo se non esiste già il
buffer
        begin
            numgara:=ABSTable14.FieldName('CODGA').AsInteger;
            if numgara<> StrToInt(Edit3.Text) then //numero gara residente DIVERSO
dal richiesto

```

```

begin
    if MessageDlg('Il Buffer contiene dati per una Gara diversa
'+#13#10+
        'e con codice : '+ IntToStr(numgara)+' Desideri cancellarli ?',
mtconfirmation,
        [mbYes,mbNo], 0) = mrYes then
        begin
            Button7Click(self);
            ABSTable14.Open;
            // Edit3.Text:='';
            // Exit;
        end else
        begin //Gara diversa da quella richiesta non cancellata
            ShowMessage('Ricorda che puoi esaminare una gara per
volta!');

            Edit3.Text:='';
            Exit;
        end;
    end else
    begin
        btscelta := MessageDlg('La Tabella per la Gara n.
'+IntToStr(numgara)+' ha '+IntToStr(quantit)+
        ' dati.'+#13+' Vuoi usarla(Yes) o cancellarla(No) o
rinunciare(Cancel) ? ',mtCustom,
            [mbYes,mbNo,mbCancel], 0);
        if btscelta = mrYes then //usa il precedente buffer e continua
        begin
            dbgrd7.Columns[0].FieldName:='NM'; //applicare il campo relativo
al cambio di Dataset
            ds7.DataSet:=ABSTable14;
            Label6.Caption:='Dati relativi ad una Gara - su buffer';
            ShowMessage('E' possibile inserire i Tempi delle discipline in
gara'+#13#10+
                'o modificare gli altri dati degli Atleti.');
```

Edit3.Text:=''; //Punto in cui si è scelta la GARA da Aggiornare
 Exit;
 end;
 if btscelta = mrNo then //cancella il buffer per rifarlo
 begin
 Button7Click(Self);
 Edit3.Text:='';
 exit;
 end;
 if btscelta = mrCancel then //esce dalla procedura
 begin
 Edit3.Text:='';
 Exit;
 end;
 end;
end;

dbgrd7.Columns[0].FieldName:='NM'; //applicare il campo relativo al cambio di
Dataset
ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
with ABSQuery7 do
begin
 Close; //porto su DBGRID solo la gara scelta con tutti gli
iscritti

```

        SQL.Text := 'Select * from Gara where CODGA =
'+QuotedStr(Edit3.Text)+ ' ORDER BY CODATL Asc, CODDSPL Asc';
        ExecSql;
        Open;           //Estrae i dati di iscrizione solo per una gara
        First;
        while (not Eof) do //sempre sotto ABSQuery7
            begin           // copia il Query in una
Table  14 Buffer gara corrente
                Form6.ABSTable14.Insert;

Form6.ABSTable14.FieldName('NM').AsInteger:=ABSQuery7.FieldName('NMREC').AsInteger;

Form6.ABSTable14.FieldName('CODGA').AsInteger:=ABSQuery7.FieldName('CODGA').AsInteger;

Form6.ABSTable14.FieldName('CODDSPL').AsInteger:=ABSQuery7.FieldName('CODDSPL').AsInteger;

Form6.ABSTable14.FieldName('CODATL').AsInteger:=ABSQuery7.FieldName('CODATL').AsInteger;

Form6.ABSTable14.FieldName('COGNOME').AsString:=ABSQuery7.FieldName('COGNOME').AsString;

Form6.ABSTable14.FieldName('NMPET').AsInteger:=ABSQuery7.FieldName('NMPET').AsInteger;

Form6.ABSTable14.FieldName('NMPOS').AsInteger:=ABSQuery7.FieldName('NMPOS').AsInteger;

Form6.ABSTable14.FieldName('TAME1').AsDateTime:=ABSQuery7.FieldName('TAME1').AsDateTime;

Form6.ABSTable14.FieldName('TBME2').AsDateTime:=ABSQuery7.FieldName('TBME2').AsDateTime;

Form6.ABSTable14.FieldName('DQ').AsBoolean:=ABSQuery7.FieldName('DQ').AsBoolean;

Form6.ABSTable14.FieldName('PEN').AsDateTime:=ABSQuery7.FieldName('PEN').AsDateTime;

Form6.ABSTable14.FieldName('MANL').AsBoolean:=ABSQuery7.FieldName('MANL').AsBoolean;

Form6.ABSTable14.FieldName('CAT').AsInteger:=ABSQuery7.FieldName('CAT').AsInteger;

                Form6.ABSTable14.Post;
                ABSQuery7.Next;
            end;
        end;
        Label6.Caption:='Dati relativi ad una Gara';
        ds7.DataSet:=ABSTable14;           //MOSTRA LA TABELLA GARA CORRENTE
        if ABSTable14.RecordCount>0 then
            begin

```



```

TDateTimeField(ABSTable14.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';

TDateTimeField(ABSTable14.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
TDateTimeField(ABSTable14.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
TBooleanField(ABSTable14.FieldByName('DQ')).DisplayValues:='Si;No';
TBooleanField(ABSTable14.FieldByName('MANL')).DisplayValues:='Si;No';
end;

//Controllo presenza dati essenziali
cosa:=TStringList.Create;
cosa.Clear;
ABSTable14.First;
while not ABSTable14.Eof do
begin
if ABSTable14.FieldByName('CAT').AsInteger=0 then
begin
ga:=ABSTable14.FieldByName('CODGA').AsInteger;
ca:=ABSTable14.FieldByName('CODATL').AsInteger;
cd:=ABSTable14.FieldByName('CODDSPL').AsInteger;
cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '
+inttostr(ca)+'- Disciplina: '+inttostr(cd)+'> Categoria');
end;
if ABSTable14.FieldByName('NMPET').AsInteger=0 then
begin
ga:=ABSTable14.FieldByName('CODGA').AsInteger;
ca:=ABSTable14.FieldByName('CODATL').AsInteger;
cd:=ABSTable14.FieldByName('CODDSPL').AsInteger;
cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '
+inttostr(ca)+'- Disciplina: '+inttostr(cd)+'> Pettorale');
end;
if ABSTable14.FieldByName('NMPOS').AsInteger=0 then
begin
ga:=ABSTable14.FieldByName('CODGA').AsInteger;
ca:=ABSTable14.FieldByName('CODATL').AsInteger;
cd:=ABSTable14.FieldByName('CODDSPL').AsInteger;
cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '
+inttostr(ca)+'- Disciplina: '+inttostr(cd)+'> Posizione');
end;
if ABSTable14.FieldByName('MANL').AsBoolean=False then
begin
ga:=ABSTable14.FieldByName('CODGA').AsInteger;
ca:=ABSTable14.FieldByName('CODATL').AsInteger;
cd:=ABSTable14.FieldByName('CODDSPL').AsInteger;
cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '
+inttostr(ca)+'- Disciplina: '+inttostr(cd)+'> Manleva');
end;
ABSTable14.Next;
end;
if cosa.Count>0 then
begin
cosa.SaveToFile('.\mancanti.txt');
if (cosa.Count<50) then
ShowMessage('Codici Atleti con dati mancanti-Annotarli ' + #13+#13+
cosa.Text+#13+
' Dati salvati in "mancanti.txt") else
ShowMessage('La lista degli Atleti con dati mancanti è salvata in
"Mancanti.txt".');

```

```

        //Correzione DATI Mancanti
if MessageDlg('Desideri fornire ora i dati di iscrizione mancanti
?'+#13+#13
        + 'Ricorda di sincronizzare',mtconfirmation, [mbYes,mbNo], 0) =
mrYes then
    begin
        Form6.ABSTable14.Close;           //Attenzione nel caso dei dati
mancanti
        Form6.ABSTable14.EmptyTable; // la T14 viene ricostruita con
solo
        Form6.ABSTable14.Open;           // i mancanti
        Form6.ABSTable14.First;
        Form6.ABSTable14.Edit;
        with ABSQuery7 do
            begin
                Close;           //inserisce solo gli atleti con dati preliminari
mancanti
                SQL.Text := 'Select * from Gara where CODGA =
'+QuotedStr(Edit3.Text)+' and (CAT='+QuotedStr(IntToStr(0))+
                ' OR NMPET='+QuotedStr(IntToStr(0))+ 'OR
NMPOS='+QuotedStr(IntToStr(0))+ 'OR MANL = False) ' ;
                ExecSql;
                Open;
                ABSQuery7.First;
                while (not Eof) do //sempre sotto ABSQuery7
                    begin // copia il Query in una
Table 14 Buffer gara corrente
                        Form6.ABSTable14.Insert;

Form6.ABSTable14.FieldName('NM').AsInteger:=ABSQuery7.FieldName('NMREC').AsInteger;

Form6.ABSTable14.FieldName('CODGA').AsInteger:=ABSQuery7.FieldName('CODGA').AsInteger;

Form6.ABSTable14.FieldName('CODDSPL').AsInteger:=ABSQuery7.FieldName('CODDSPL').AsInteger;

Form6.ABSTable14.FieldName('CODATL').AsInteger:=ABSQuery7.FieldName('CODATL').AsInteger;

Form6.ABSTable14.FieldName('COGNOME').AsString:=ABSQuery7.FieldName('COGNOME').AsString;

Form6.ABSTable14.FieldName('NMPET').AsInteger:=ABSQuery7.FieldName('NMPET').AsInteger;

Form6.ABSTable14.FieldName('NMPOS').AsInteger:=ABSQuery7.FieldName('NMPOS').AsInteger;

Form6.ABSTable14.FieldName('TAME1').AsDateTime:=ABSQuery7.FieldName('TAME1').AsDateTime;

Form6.ABSTable14.FieldName('TBME2').AsDateTime:=ABSQuery7.FieldName('TBME2').AsDateTime;

Form6.ABSTable14.FieldName('DQ').AsBoolean:=ABSQuery7.FieldName('DQ').AsBoolean;

```

```

Form6.ABSTable14.FieldName('PEN').AsDateTime:=ABSQuery7.FieldName('PEN').AsDateTime;

Form6.ABSTable14.FieldName('MANL').AsBoolean:=ABSQuery7.FieldName('MANL').AsBoolean;

Form6.ABSTable14.FieldName('CAT').AsInteger:=ABSQuery7.FieldName('CAT').AsInteger;

        Form6.ABSTable14.Post;
        ABSQuery7.Next;
    end;
    ds7.DataSet:=ABSTable14;
end;
    Form10.Show;
    if Form6.ABSTable14.RecordCount>0 then
        mncti:=True else mncti:=False;
    end;
end;
if mncti=False then
    ShowMessage('E' possibile inserire i Tempi delle discipline di Gara'+#13#10+'o modificare gli altri dati degli Atleti.');
```

Edit3.Text:='';
cosa.Clear;

end;

procedure TForm6.Button7Click(Sender: TObject); *// Resetta la tabella della Gara corrente*
begin
 if MessageDlg('Stai per azzerare i dati memorizzati nel Buffer. Vuoi continuare?',
 mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes **then**
 begin
 ABSTable14.Close;
 ABSTable14.EmptyTable;
 ShowMessage('Tutti i dati di iscrizione contenuti nel buffer sono stati cancellati,'
 +#13+' i dati sincronizzati sono presenti nel Database');

end;
 ds7.DataSet:=ABSTable14;
end;

procedure TForm6.Button8Click(Sender: TObject); *//Sincronizza buffer con Gara*
var
doppio:Boolean;
cerco:integer;
begin
 if filtGa=False **then**
 begin
 ShowMessage('Non c'è nulla da sincronizzare.');

 Exit;
 end;
 if MessageDlg('Confermi di voler aggiornare i Dati dello Storico?',
mtconfirmation,
 [mbYes,mbNo], 0) = mrNo **then** Exit;

```

Form2.ABSTable7.Open;
ABSTable14.Open;
ABSTable14.First;
ABSTable14.Edit;
while not ABSTable14.eof do
  begin
    cerco:=ABSTable14.FieldName('NM').AsInteger;
    doppio:=Form2.ABSTable7.Locate('NMREC',cerco,[]);
    if doppio=True then
      begin //record preesistente da aggiornare
        Form2.ABSTable7.Edit;

Form2.ABSTable7.FieldName('NMPET').AsInteger:=ABSTable14.FieldName('NMPET')
).AsInteger;

Form2.ABSTable7.FieldName('NMPOS').AsInteger:=ABSTable14.FieldName('NMPOS'
).AsInteger;

Form2.ABSTable7.FieldName('TAME1').AsDateTime:=ABSTable14.FieldName('TAME1
').AsDateTime;

Form2.ABSTable7.FieldName('TBME2').AsDateTime:=ABSTable14.FieldName('TBME2
').AsDateTime;

Form2.ABSTable7.FieldName('DQ').AsBoolean:=ABSTable14.FieldName('DQ').AsBo
olean;

Form2.ABSTable7.FieldName('PEN').AsDateTime:=ABSTable14.FieldName('PEN').A
sDateTime;

Form2.ABSTable7.FieldName('MANL').AsBoolean:=ABSTable14.FieldName('MANL').
AsBoolean;

Form2.ABSTable7.FieldName('CAT').AsInteger:=ABSTable14.FieldName('CAT').As
Integer;
      Form2.ABSTable7.Post;
    end else
      begin
        ShowMessage('ERRORE ! Il record con il codice '+IntToStr(cerco)+
          ' non esiste nella Tabella Gare Generale.');
        Exit;
      end;
      ABSTable14.Next;
    end;
    inserisco7:=False ;
    Form9.Hide;
    Checkbox2.Checked:=False;
    Form10.Hide;
    ShowMessage('La tabella Gare Generale è stata aggiornata.');
    Button5.Click;
  end;

procedure TForm6.Button9Click(Sender: TObject);
begin
  if filtGa=False then
    begin
      ShowMessage('Seleziona una Gara da aggiornare');
      Exit;
    end;

```

```

        end;
checkbox2.Checked:=False;
Form10.Hide;
Form9.Show;
Form9.RadioButton1.Checked:=True;
end;

```

```

procedure TForm6.Button10Click(Sender: TObject); //Rapporto iscritti
begin
    if Edit3.Text='' then
        begin
            ShowMessage('Inserisci il codice della Gara nella casella a destra !');
            Exit;
        end;

        frxReport14.ShowReport;

```

```

end;

```

```

procedure TForm6.Button11Click(Sender: TObject); //Reset
begin
    Edit1.Text:='';
    Edit2.Text:='';
    Edit4.Text:='';
    cbb74.Text:='Scegli';
    rb1.Checked:=True;
end;

```

//Inserimento atleti nella gara cliccando su un Atleta

```

procedure TForm6.dbgrd72CellClick(Column: TColumn);
var
    Nom,Cogn:string;
    volte:smallInt;
    Sorge:TABSTable;
    Rect2:Boolean;
    trovato:Boolean;
    n:Integer;
begin
    if Edit1.Text<>' ' then
        begin
            // o usa la Tab2=ATL o Query2 = ATL da cognome
            Form2.ABSTable2.Open;
            if scelta=False then
                begin
                    ATL:= dbgrd72.Fields[0].AsInteger;//CODATL dal campo 0 della tabella di cui
ho cliccato una colonna
                    end else ATL:=AbsQuery72.FieldByName('CODATL').AsInteger; //prende ATL
dalla scelta

                    if Form2.ABSTable2.RecordCount >0 then
                        Rect2:=Form2.ABSTable2.Locate('CODATL',ATL,[]) else
                            begin
                                ShowMessage('La tabella degli Atleti è vuota. ');
                                Exit;
                            end;
                        if Rect2=False then
                            begin

```

```

        ShowMessage('Errore Imprevisto: l''Atleta non esiste nel DataBase -
Controlla');
        Exit;
    end;
    Cogn:= Form2.ABSTable2.FieldName('COGNOME').AsString;
    Nom:= Form2.ABSTable2.FieldName('NOME').AsString;

    if rb1.Checked=True then //automatica
    begin
        if MessageDlg('Confermi l''iscrizione automatica ?',
            mtconfirmation, [mbYes,mbNo], 0) = mrYes then
            begin
                if (cbb74.ItemIndex=-1) OR (cbb74.Text='Scegli') then //Att.
cbb.Text='xx' funge dopo una scelta
                begin
                    ShowMessage('Scegli il Gruppo di iscrizione');
                    Exit;
                end;
            end;
        //SCELTA AUTOMATICA dopo la scelta della Gara e del Gruppo
        if MessageDlg('Confermi l''iscrizione AUTOMATICA di '+Nom+ ' '+ Cogn+
            ' alle discipline del gruppo '+cbb74.Text,
            mtconfirmation, [mbYes,mbNo], 0) = mrNo then
        begin
            Button2.Click;
            Exit;
        end else
        begin
            if cbb74.Text='Base' then
            begin
                Sorge:=Form2.ABSTable10;
                Sorge.Open;
            end;

            if cbb74.Text='GruppoA' then
            begin
                Sorge:=Form2.ABSTable11;
                Sorge.Open;
            end;

            if cbb74.Text='GruppoB' then
            begin
                Sorge:=Form2.ABSTable12;
                Sorge.Open;
            end;
            //CONTROLLO CATEGORIA
            if Edit4.Text='' then
            begin
                ShowMessage(' Inserisci la Categoria');
                Exit;
            end;
            //Verifica se esiste già l'iscrizione
            Sorge.First;
            while not Sorge.Eof do
            begin
                n:=Sorge.FieldName('CODDSPL').AsInteger;

                trovato:=Form2.ABSTable7.Locate('CODGA;CODDSPL;CODATL',VarArrayOf([Edit1.Text,

```

```

IntToStr(n), IntToStr(ATL) ]), []);
    if trovato=True then
        begin
            ShowMessage('ATTENZIONE ! - Questo Atleta risulta già iscritto
,
            + #13 + 'ad una o più gare del Gruppo. Non è consentita
l''iscrizione automatica');
            Exit;
        end;
        Sorge.Next;
    end;

    //Carica il nuovo record
    Sorge.First;
    while not Sorge.EOF do // Modo corretto di loop EOF
    begin
        Form2.ABSTable7.Insert;

Form2.ABSTable7.FieldByName('CODGA').AsInteger:=StrToInt(Edit1.Text);

Form2.ABSTable7.FieldByName('CODDSPL').AsInteger:=Sorge.FieldByName('CODDSPL')
.AsInteger;
        Form2.ABSTable7.FieldByName('CODATL').AsInteger:= ATL;
        Form2.ABSTable7.FieldByName('COGNOME').AsString:=Cogn;
        Form2.ABSTable7.FieldByName('NMPET').AsInteger:=0;
        Form2.ABSTable7.FieldByName('NMPOS').AsInteger:=0;

Form2.ABSTable7.FieldByName('TAME1').AsDateTime:=StrToTime('00:00:00');

Form2.ABSTable7.FieldByName('TBME2').AsDateTime:=StrToTime('00:00:00');
        Form2.ABSTable7.FieldByName('DQ').AsBoolean:=False;

Form2.ABSTable7.FieldByName('PEN').AsDateTime:=StrToTime('00:00:00,000');
        Form2.ABSTable7.FieldByName('MANL').AsBoolean:=False;

Form2.ABSTable7.FieldByName('CAT').AsInteger:=StrToInt(Edit4.Text);
        Form2.ABSTable7.Post; //Carica il dato di iscrizione aggiungendo
un record
        Sorge.Next;
    end;
    //Edit1.Text:='';
    Edit4.Text:='';
    //cbb74.Text:='Scegli';
    Button5.Click;
end; //Iscrizione automatica terminata
end;

//SCELTA MANUALE

if (rb2.Checked=True) and (Edit2.Text<>'') then //verifica se
già inserito
begin
    ds7.DataSet:=ABSQuery7;
    with AbsQuery7 do
    begin
        Close; //cerca la coerenza gare - discipline
        SQL.Text := 'Select * from Gara where CODDSPL =
'+QuotedStr(Edit2.Text)+

```

```

'AND CODGA =' + QuotedStr(Edit1.Text);
ExecSql;
Open;
volte:=AbsQuery7.RecordCount;
if volte=0 then
    begin
        Application.MessageBox('I dati di Gara e di Disciplina non sono
compatibili.'
        + #13#10 +
        'Nella procedura di iscrizione manuale, il Sistema' +
        #13#10 + 'non ne controlla la correttezza.', 'Attenzione',
        MB_OK + MB_ICONWARNING);

        Edit1.Text:='';
        Edit2.Text:='';
        Edit4.Text:='';
        rb1.Checked:=True;
        Exit;
    end;
end;

    with AbsQuery7 do
    begin
        Close; //cerca le doppie iscrizioni
        SQL.Text := 'Select * from Gara where CODATL = ' +
QuotedStr(IntToStr(ATL)) +
        ' AND CODDSPL = ' + QuotedStr(Edit2.Text) + 'AND CODGA
=' + QuotedStr(Edit1.Text);
        ExecSql;
        Open;
        volte:=AbsQuery7.RecordCount;
        if volte>0 then
            begin
                ShowMessage(Nom+ ' ' + Cogn+ ' è già iscritto ' + IntToStr(volte) + '
volte a questa disciplina');
                Button5.Click;
                Form2.ABSTable2.First;
                Exit;
            end;
        end;
        if Edit4.Text='' then
            begin
                ShowMessage(' Inserisci la Categoria');
                Exit;
            end;
        if MessageDlg('Confermi l''iscrizione di ' + Nom+ ' ' + Cogn,
mtconfirmation, [mbYes,mbNo], 0) = mrYes then
            begin
                Form2.ABSTable7.Insert;
                Form2.ABSTable7.FieldName('CODATL').AsInteger:=ATL;
                Form2.ABSTable7.FieldName('CODGA').AsInteger:=
StrToInt(Edit1.Text);

                Form2.ABSTable7.FieldName('CODDSPL').AsInteger:=StrToInt(Edit2.Text);
                Form2.ABSTable7.FieldName('COGNOME').AsString:= Cogn;
                Form2.ABSTable7.FieldName('NMPET').AsInteger:=0;
                Form2.ABSTable7.FieldName('NMPOS').AsInteger:=0;

```



```

Form2.ABSTable7.FieldByName('TAME1').AsDateTime:=StrToTime('00:00:00');

Form2.ABSTable7.FieldByName('TBME2').AsDateTime:=StrToTime('00:00:00');
Form2.ABSTable7.FieldByName('DQ').AsBoolean:=False;

Form2.ABSTable7.FieldByName('PEN').AsDateTime:=StrToTime('00:00:00,000');
Form2.ABSTable7.FieldByName('MANL').AsBoolean:=False;

Form2.ABSTable7.FieldByName('CAT').AsInteger:=StrToInt(Edit4.Text);
Form2.ABSTable7.Post;
Button5.Click; //per ripristinare l'evento si ds72
Form2.ABSTable2.First;
Edit1.Text:='';
Edit2.Text:='';
Edit4.Text:='';
cbb74.Text:='Scegli';
inserisco7:=False;
end else Exit;
end else
if rb2.Checked=True then
begin
ShowMessage('Iscrizione manuale, indica prima il codice
disciplina ');
Button2.Click;
exit;
end;
end else
begin
ShowMessage('Inserisci il Codice della Gara. ');
Exit;
end;
end;

//INSERISCE I Tempi (da unit Tempi o Mancanti)
procedure TForm6.dbgrd7CellClick(Column: TColumn); //Cliccare sulla cella della
DBGrid7
var
catl,cga:smallint;
begin
if FiltGa=False then
begin
ShowMessage('Non si può editare lo storico.'+#13+'Inserisci prima il codice
Gara da editare') ;
Exit;
end else
begin

if inserisco7=True then Exit; //quando ho già inserito
dbgrd7.ReadOnly:=False; //tempo è caricato da Temp con i dati che si vogliono
trasferire
if (tempo='') and (checkbox1.Checked=False) and (mncti=False) then
begin
ShowMessage('Per inserire i dati utilizzare "Tempi IN" e "DQ IN"');
Exit;
end ;
if (tempo<>'') and (checkbox1.Checked=False) then //Inserimento Tempi della
disciplina
begin

```

```

ABSTable14.Edit; //buffer gara corrente

TDateTimeField(ABSTable14.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';

TDateTimeField(ABSTable14.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
TDateTimeField(ABSTable14.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
TBooleanField(Form6.ABSTable14.FieldByName('DQ')).DisplayValues:='Si;No';

TBooleanField(Form6.ABSTable14.FieldByName('MANL')).DisplayValues:='Si;No';
  if (rab1 = True) then //rab1 èTempoA
    begin
      if ABSTable14.FieldByName('TAME1').AsDateTime<>0 then
        if MessageDlg('Esiste giàun dato, vuoi sostituirlo?',
          mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
        ABSTable14.FieldByName('TAME1').AsDateTime:=StrToTime(tempo);
      end;
    if (rab2 = True) then //rab2 èTempoB
      begin
        if ABSTable14.FieldByName('TBME2').AsDateTime<>0 then
          if MessageDlg('Esiste giàun dato, vuoi sostituirlo?',
            mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
          ABSTable14.FieldByName('TBME2').AsDateTime:=StrToTime(tempo);
        end;
      if (rab3 = True) then //rab3 èPenalità
        begin
          if ABSTable14.FieldByName('PEN').AsDateTime<>0 then
            if MessageDlg('Esiste giàun dato, vuoi sostituirlo?',
              mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
            ABSTable14.FieldByName('PEN').AsDateTime:=StrToTime(tempo);
          end;
        ABSTable14.Post;
        Inserisco7:=True;
        Form9.Edit1.text:='';
        Form9.Edit2.Text:='';
        Form9.Edit3.Text:='';
        Form9.RadioButton1.Checked:=True;
        DBGrd7.DataSource.DataSet.Refresh;
        tempo:='';
      end;

    if (tempo='') and (checkbox1.Checked) then //per inserire il DQ
      begin
        dbgrd7.ReadOnly:=False;
        ABSTable14.Edit; //buffer gara corrente
        if ABSTable14.FieldByName('DQ').AsBoolean=True then
          begin
            if MessageDlg('Esiste giàun DQ, vuoi cancellarlo?',
              mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrYes then
              ABSTable14.FieldByName('DQ').AsBoolean:=False
            end else
              ABSTable14.FieldByName('DQ').AsBoolean:=True;
            checkbox1.Checked:=False;
            ABSTable14.Post;
            Inserisco7:=True;
            DBGrd7.DataSource.DataSet.Refresh;
          end;
        if (mancante>0) AND (mncat=True) then //per inserire un mancante CAT

```

```

begin
    dbgrd7.ReadOnly:=False;
    ABSTable14.Edit; //buffer gara corrente
    if ABSTable14.FieldName('CAT').AsInteger>0 then

        if MessageDlg('Esiste giàuna Categoria, vuoi sostituirla ?',
            mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
        ABSTable14.FieldName('CAT').AsInteger:=mancante;
        ABSTable14.Post;
        Mancante:=0;
        mncat:=False;
        Form10.Edit1.Text:='';
        DBGrd7.DataSource.DataSet.Refresh;
    end;

    if (mancante>0) AND (mnpet=True ) then //per inserire un mancante NMPET
    begin
        dbgrd7.ReadOnly:=False;
        ABSTable14.Edit; //buffer gara corrente
        if ABSTable14.FieldName('NMPET').AsInteger>0 then

            if MessageDlg('Esiste giàun numero di Pettorale, vuoi sostituirlo ?',
                mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit ;
            ABSTable14.FieldName('NMPET').AsInteger:=mancante;
            ABSTable14.Post;
            catl:=ABSTable14.FieldName('CODATL').AsInteger;
            cga:=ABSTable14.FieldName('CODGA').AsInteger;
            if MessageDlg('Desideri applicare lo stesso numero di Pettorale per
,
                + #13#10 +
                'tutte le Discipline a cui questo Atleta partecipa ?',
                mtConfirmation, [mbYes, mbNo], 0)=mrYes then
            begin
                ABSTable14.First;
                while (not ABSTable14.EOF) do
                begin
                    ABSTable14.Edit; //per essere certi dell'editmode
                    if (ABSTable14.FieldName('CODATL').AsInteger = catl)
and
                        (ABSTable14.FieldName('CODGA').AsInteger = cga) then
                        begin
ABSTable14.FieldName('NMPET').AsInteger:=mancante;
                            ABSTable14.post;
                            end;
                            ABSTable14.next; //di regola crea un post automatico
                        end; //ma toglie da edit mode
                            ABSTable14.Next;MessageDlg('In questa Gara, questo Atleta
ha ricevuto'
                                + #13#10 +
                                'un unico numero di Pettorale. SINCRONIZZA',
                                mtWarning, [mbOK], 0);
                        end;
                            Form10.Edit1.Text:='';
                            Mancante:=0;
                            mnpet:=False;
                            DBGrd7.DataSource.DataSet.Refresh;
                        end;

```

```

    if (mancante>0) AND (mnpos=True ) then      //per inserire un mancante NMPOS
begin
    dbgrd7.ReadOnly:=False;
    ABSTable14.Edit; //buffer gara corrente
    if ABSTable14.FieldName('NMPOS').AsInteger>0 then

        if MessageDlg('Esiste già un numero di Posizione, vuoi sostituirlo ?',
            mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit ;
        ABSTable14.FieldName('NMPOS').AsInteger:=mancante;
        ABSTable14.Post;
        catl:=ABSTable14.FieldName('CODATL').AsInteger;
        cga:=ABSTable14.FieldName('CODGA').AsInteger;
        if MessageDlg('Desideri applicare lo stesso numero di Posizione per
,
        + #13#10 +
        'tutte le Discipline a cui questo Atleta partecipa ?',
        mtConfirmation, [mbYes, mbNo], 0)=mrYes then
        begin
            ABSTable14.First;
            while (not ABSTable14.EOF) do
            begin
                ABSTable14.Edit;
                if (ABSTable14.FieldName('CODATL').AsInteger = catl)

and
                (ABSTable14.FieldName('CODGA').AsInteger = cga) then
                begin
                    ABSTable14.FieldName('NMPOS').AsInteger:=mancante;
                    ABSTable14.post;
                    end;
                ABSTable14.next;
            end;
            ABSTable14.Next;MessageDlg('In questa Gara, questo Atleta
ha ricevuto'

            + #13#10 +
            'un unico numero di Posizione. SINCRONIZZA',
            mtWarning, [mbOK], 0);

        end;
        Mancante:=0;
        mnpos:=False;
        Form10.Edit1.Text:='';
        DBGrd7.DataSource.DataSet.Refresh;
    end;

    if (mnmanl=True ) then      //per inserire la Manleva
begin
    dbgrd7.ReadOnly:=False;
    ABSTable14.Edit; //buffer gara corrente
    if ABSTable14.FieldName('MANL').AsBoolean=True then
        if MessageDlg('Esiste già la Manleva, vuoi Cancellarla ?',
            mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit ;
        ABSTable14.FieldName('MANL').AsBoolean:=manleva;
        ABSTable14.Post;
        catl:=ABSTable14.FieldName('CODATL').AsInteger;
        cga:=ABSTable14.FieldName('CODGA').AsInteger;
        if MessageDlg('Desideri registrare la Manleva per '
        + #13#10 +
        'tutte le Discipline a cui questo Atleta partecipa ?',

```

```

mtConfirmation, [mbYes, mbNo], 0)=mrYes then
begin
  ABSTable14.First;
  while (not ABSTable14.EOF) do
    begin
      ABSTable14.Edit;
      if (ABSTable14.FieldName('CODATL').AsInteger = catl)
and
      (ABSTable14.FieldName('CODGA').AsInteger = cga) then
        begin
          ABSTable14.FieldName('MANL').AsBoolean:=manleva;
          ABSTable14.post;
          end;
          ABSTable14.next;
        end;
        ABSTable14.Next;MessageDlg('In questa Gara, questo Atleta
ha registrato'
          + #13#10 +
          'la Manleva in tutte le Discipline. SINCRONIZZA',
          mtWarning, [mbOK], 0);
        end;
        mnmanl:=False;
        Form10.Edit1.Text:='';
        DBGrd7.DataSource.DataSet.Refresh;
      end;
      //trucco per poter fare l'onclick sulla stessa riga due volte
      Form6.dbgrd7.SelectedField:=ABSTable14.FieldName('NM');
    end;
  end;

  // Ricerca per cognome

  // seleziona una lettera dalla combobox per poi inserire l'atleta
procedure TForm6.cbb72Select(Sender: TObject);
var
c:string;
begin
  if Edit1.Text<>' ' then
    begin
      c:=cbb72.Items[cbb72.ItemIndex]; //estrae la scelta dalla combobox
      scelta:=True; //scelta fatta per cognome
      ds72.DataSet:=AbsQuery72;
      with AbsQuery72 do
        begin
          Close; //inserisce una variabile nel query selezionandola per primo
carattere
          SQL.Text := 'Select * from AnaAtl where substring(COGNOME,1,1) = '+
QuotedStr(c);
          ExecSql;
          Open;
          cbb72.ItemIndex:=-1;
        end;
      end else
        begin
          cbb72.ItemIndex:=-1; //riporta la combobox all'inizio
          ShowMessage('Inserisci il Codice della Gara');
          Exit;

```

```

    end;
end;

procedure TForm6.cbb74Change(Sender: TObject); //SCEGLIE IL GRUPPO
var //realizza per ogni gruppo scelto una tabella delle DSPL per quella gara
    dsplSi:Boolean;
    DSPLn:Integer;
    baseSi:Boolean;
begin
    Form2.ABSTable6.Open;
    Form2.ABSTable10.TableName:='repBase';
    Form2.ABSTable10.Open;
    Form2.ABSTable10.Edit;
    Form2.ABSTable11.TableName:='repGRPA';
    Form2.ABSTable11.Open;
    Form2.ABSTable11.Edit;
    Form2.ABSTable12.TableName:='repGRPB';
    Form2.ABSTable12.Open;
    Form2.ABSTable12.Edit;
    if Edit1.Text<>' ' then
    begin
        if cbb74.Text='Base' then // GRUPPO BASE
        begin
            if Form2.ABSTable6.RecordCount>0 then //Controlla se per la Gara scelta
l'accoppiamento di Base
                begin // esiste già
                    baseSi:=
Form2.ABSTable6.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'esistenza
                    if baseSi=True then //del codice della Gara negli
accoppiamenti di Base
                        begin
                            if MessageDlg('Il Gruppo delle discipline Base di questa
Gara'+#13+
                                'è già stato creato. Vuoi rifarlo ?', mtconfirmation,
[mbYes,mbNo], 0,mbNo) = mrYes then
                                begin
                                    ShowMessage('Esci dalla Sezione Gara Corrente e lancia
Anagrafica Gare'+#13+
                                        'per rifare gli accoppiamenti Gruppo Base');
                                    Edit1.Text:='';
                                    Edit4.Text:='';
                                    Exit;
                                end;
                            end else
                            begin
                                ShowMessage('Mancano accoppiamenti o manca la Gara
scelta.'+#13+
                                    'Lancia Anagrafica Gare per effettuare un controllo');
                                Edit1.Text:='';
                                Edit4.Text:='';
                                Exit;
                            end;
                        end;
                    end;
                    Form2.ABSTable10.Close;
                    Form2.ABSTable10.EmptyTable;
                    ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo
le discipline

```

```

        with ABSQuery74 do                // del Gruppo Base di quella Gara
        begin
            Close; //estrae le discipline relative alla gara ed al Gruppo Base
            SQL.Text := 'Select * from Base where CODGA =
'+QuotedStr(edit1.Text);
            ExecSql;
            Open; //la Tabella è formata da tutti i codici disciplina della
Base
            First;
            Form2.ABSTable10.Open;
            Form2.ABSTable10.Edit;
            Form2.ABSTable4.Open;
            while(not Eof)do //Crea le tabelle complete delle discipline per
quel Gruppo
            begin
                DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della
DSPL
                dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod
per localizzare il record di ABS/4
                if dsplSi=True then
                begin
                    Form2.ABSTable10.Insert; //Costruisce la tabella completa
delle discipline
                    Form2.ABSTable10.FieldByName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldByName(
('CODDSPL').AsInteger;
                    Form2.ABSTable10.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName('D
ESCR').AsString;
                    Form2.ABSTable10.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('
TMAX').AsDateTime;
                    Form2.ABSTable10.Post
                end else
                begin
                    ShowMessage('Attenzione ! Manca questo Codice dalla
Tabella delle Discipline : '
                    + IntToStr(DSPLn));
                    Exit;
                end;
                Next;
            end;

            ds74.DataSet:=Form2.ABSTable10;
        end;

        ShowMessage('La tabella mostra le discipline del Gruppo Base'+#13+
' per la Gara indicata.');
```

end;

```

        if cbb74.Text='GruppoA' then
        begin
            Form2.ABSTable8.Open;
            if Form2.ABSTable8.RecordCount>0 then //Controlla se per la Gara scelta
l'accoppiamento di GruppoA
            begin
                baseSi:= // esiste già
                Form2.ABSTable8.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'esistenza

```

```

        if baseSi=True then           //del codice della Gara negli
accoppiamenti di Base
        begin
            if MessageDlg('Il Gruppo A delle discipline di questa
Gara'+#13+
                'è già stato creato. Vuoi rifarlo ?', mtconfirmation,
[mbYes,mbNo], 0,mbNo) = mrYes then
                begin
                    ShowMessage('Esci dalla Sezione Gara Corrente e lancia
Anagrafica Gare'+#13+
                        'per rifare gli accoppiamenti del Gruppo A');
                    Exit;
                end;
            end else
            begin
                ShowMessage('Esci dalla Sezione Gara Corrente e lancia
Anagrafica Gare'+#13+
                    'per definire gli accoppiamenti Gruppo A');
                Exit;
            end;
        end;

        Form2.ABSTable11.Close;
        Form2.ABSTable11.EmptyTable;
        ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo
le discipline
        with ABSQuery74 do           // del Gruppo A di quella Gara
        begin
            Close; //estrae le discipline relative alla gara ed al Gruppo A
            SQL.Text := 'Select * from GRPA where CODGA =
'+QuotedStr(edit1.Text);
            ExecSql;
            Open; //la Tabella è formata da tutti i codici disciplina della
Base

            First;
            Form2.ABSTable11.Open;
            Form2.ABSTable11.Edit;
            while(not Eof)do //Crea le tabelle complete delle discipline per
quel Gruppo
            begin
                DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della
DSPL

                dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod
per localizzare il record di ABS/4
                if dsplSi=True then
                begin
                    Form2.ABSTable11.Insert; //Costruisce la tabella completa
delle discipline

                    Form2.ABSTable11.FieldName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldName(
'CODDSPL').AsInteger;

                    Form2.ABSTable11.FieldName('DESCR').AsString:=Form2.ABSTable4.FieldName('D
ESCR').AsString;

                    Form2.ABSTable11.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName('
TMAX').AsDateTime;

                    Form2.ABSTable11.Post

```



```

        end else
            begin
                ShowMessage('Attenzione ! Manca questo Codice dalla
Tabella delle Discipline : '
                + IntToStr(DSPLn));
                Exit;
            end;
        Next;
    end;

    ds74.DataSet:=Form2.ABSTable11;
end;

    ShowMessage('La tabella mostra le discipline del Gruppo A');
end;

    if cbb74.Text='GruppoB' then
    begin
        Form2.ABSTable9.Open;
        if Form2.ABSTable9.RecordCount>0 then //Controlla se per la Gara scelta
l'accoppiamento di Gruppo B
            begin
                // esiste già
                baseSi:=
Form2.ABSTable9.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'esistenza
                if baseSi=True then //del codice della Gara negli accoppiamenti
di Base
                    begin
                        if MessageDlg('Il Gruppo B delle discipline di questa
Gara'+#13+
                        'è già stato creato. Vuoi rifarlo ?', mtconfirmation,
[mbYes,mbNo], 0,mbNo) = mrYes then
                            begin
                                ShowMessage('Esci dalla Sezione Gara Corrente e lancia
Anagrafica Gare'+#13+
                                'per rifare gli accoppiamenti del Gruppo B');
                                Exit ;
                            end;
                        end else
                        begin
                            ShowMessage('Esci dalla Sezione Gara Corrente e lancia
Anagrafica Gare'+#13+
                            'per definire gli accoppiamenti del Gruppo B');
                            Exit ;
                        end;
                    end;
                Form2.ABSTable12.Close;
                Form2.ABSTable12.EmptyTable;
                ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo
le discipline
                with ABSQuery74 do // del Gruppo B di quella Gara
                begin
                    Close; //estrae le discipline relative alla gara ed al Gruppo B
                    SQL.Text := 'Select * from GRPB where CODGA =
'+QuotedStr(edit1.Text);
                    ExecSql;
                    Open; //la Tabella è formata da tutti i codici disciplina della
Base
                    First;

```

```

        Form2.ABSTable12.Open;
        Form2.ABSTable12.Edit;
        while(not Eof)do //Crea le tabelle complete delle discipline per
quel Gruppo
            begin
                DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della
DSPL
                dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod
per localizzare il record di ABS/4
                if dsplSi=True then
                    begin
                        Form2.ABSTable12.Insert; //Costruisce la tabella completa
delle discipline

Form2.ABSTable12.FieldByName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldByName(
'CODDSPL').AsInteger;

Form2.ABSTable12.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName('D
ESCR').AsString;

Form2.ABSTable12.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('
TMAX').AsDateTime;
                        Form2.ABSTable12.Post
                    end else
                        begin
                            ShowMessage('Attenzione ! Manca questo Codice dalla
Tabella delle Discipline : '
                                + IntToStr(DSPLn));
                            Exit;
                        end;
                    Next;
                end;

                ds74.DataSet:=Form2.ABSTable12;
            end;
            ShowMessage('La tabella mostra le discipline del Gruppo B');
        end;
    end else
        begin
            ShowMessage('Inserisci il Codice della Gara ');
            cbb74.ItemIndex:=0;
            Exit;
        end;;
    end;

procedure TForm6.CheckBox1Click(Sender: TObject); //inserimento DQ
begin
    if FiltGa=False then
        begin
            ShowMessage('Non si puòeditare lo storico.'+#13+'Inserisci prima il codice
Gara da editare') ;
            checkBox1.OnClick:=nil; //disabilito l'evento onclick
            checkBox1.Checked:=False; //cancello l'apice nel box
            checkBox1.OnClick:=CheckBox1Click; //riabilito l'evento on click
            Exit;
        end;
    if checkBox1.Checked = True then
        begin

```

```

        if MessageDlg('Desideri inserire un DQ ?', mtconfirmation, [mbYes,mbNo],
0,mbNo) = mrYes then
            ShowMessage('Clicca sul nome dell''Atleta')else
            begin
                checkbox1.Checked:=False;
                Exit;
            end;

        end;

    end;

end;

procedure TForm6.CheckBox2Click(Sender: TObject); // CAT-PET-POS-MANL edit
begin
    if FiltGa=False then
        begin
            ShowMessage('Non si puòeditare lo storico.'+#13+'Inserisci prima il codice
Gara da editare') ;
            checkBox2.OnClick:=nil; //disabilito l'evento onclick
            checkbox2.Checked:=False; //cancello l'apice nel box
            checkBox2.OnClick:=CheckBox2Click; //riabilito l'evento on click
            Exit;
        end;
    if checkbox2.Checked = True then
        begin
            if MessageDlg('Desideri modificare i dati'+#13#10+
'CAT o PET o POS o MANL ?', mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrYes
then
                begin
                    Form10.Show;
                    mncti:=True;
                    MessageDlg('Qualsiasi modifica o dato venga inserito,' + #13#10 +
'ricorda di SINCRONIZZARE al termine,' + #13#10 +
'altrimenti il dato va perso.', mtInformation, [mbOK], 0);
                end else
                begin
                    checkbox2.Checked:=False;
                    Exit;
                end;

            end;

        end;

    end;

end;

//NAVIGATORE iscrizioni

procedure TForm6.dbnvgr7BeforeAction(Sender: TObject; Button: TNavigateBtn);
begin
    if Button = nbDelete then
        begin
            if filtGa=False then
                if MessageDlg('ATTENZIONE - stai per rimuovere un record dallo Storico Gare',
mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then
                    begin
                        Abort;
                        Exit;
                    end;

                end;

            end;

        end;

```

end;

procedure TForm6.dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);

begin

case button **of** *// DA VERIFICARE FORSE INUTILI*

 NBDELETE: *//se cancella il record - serve anche per evitare di entrare in Query*

 inserisco7:=False;

end;

if FiltGa=True **then**

 DBNvgr7.Controls[Ord(nbDelete)].Enabled:=False;

end;

procedure TForm6.Edit1Change(Sender: TObject);

begin

if rb1.Checked = True **then**

 cbb74.ItemIndex:=0;

end;

end.

```
unit Classifiche;           //05/02/2016
  //in questa unità vi sono righe di codice non utilizzate ma come commenti
interface
```

uses

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes,
Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.CheckLst,
Vcl.Grids, System.DateUtils, Data.DB, Vcl.DBGrids, Vcl.DBCtrls, Vcl.ExtCtrls,
ABSMMain, frxClass, frxDBSet, Datasnap.DBClient, Datasnap.Provider;
```

type

```
TForm8 = class (TForm)
  Esci: TButton;
  DBGrid8: TDBGrid;
  DataSource8: TDataSource;
  Button1: TButton;
  Button2: TButton;
  Button3: TButton;
  Label1: TLabel;
  ListBox8: TListBox;
  Button4: TButton;
  Button6: TButton;
  Shape1: TShape;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Shape2: TShape;
  Shape3: TShape;
  Button7: TButton;
  RadioButton1: TRadioButton;
  RadioButton2: TRadioButton;
  RadioButton3: TRadioButton;
  ComboBox1: TComboBox;
  Label6: TLabel;
  ABSQuery80: TABSQuery;
  DataSourceQry: TDataSource;
  ABSTable15: TABSTable;
  ABSQuery85: TABSQuery;
  frxReport8: TfrxReport;
  frxDBDataset85: TfrxDBDataset;
  ABSQuery83: TABSQuery;
  ABSTable16: TABSTable;
  DataSourceQry2: TDataSource;
  ABSTable18: TABSTable;
  frxReport8D: TfrxReport;
  frxDBDataset8D: TfrxDBDataset;
  frxReport16: TfrxReport;
  frxDBDataset16: TfrxDBDataset;
  frxReport16T: TfrxReport;
  frxDBDataset16T: TfrxDBDataset;
  frxReport16T2: TfrxReport;
  frxDBDataset16t2: TfrxDBDataset;
  ABSQueryRep: TABSQuery;
  ABSTable19: TABSTable;
  ABSTable16R: TABSTable;
  frxReportTot: TfrxReport;
  frxDBDatasetTot: TfrxDBDataset;
```

```

Bevel1: TBevel;
frxReportR: TfrxReport;
frxDBDatasetR: TfrxDBDataset;
ABSQueryR: TABSQuery;
Label2: TLabel;
ABSTable20: TABSTable;
frxReportCompl: TfrxReport;
procedure FormCreate(Sender: TObject);
procedure EsciClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure DBGrid8CellClick(Column: TColumn);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Calcola(Sender: TObject);
procedure Button7Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form8: TForm8;                                //ArGare Gare selezionate
    ClassFlg:Boolean=True;                        //ArDspl Discipline
    selezionate
    ArGare,ArDspl,ArCat:array of Integer; //ArCat Categorie selezionate
    SelGare:Boolean=False;
    SelDspl:Boolean=False;
    ClassDp:Boolean=False;
    idx,mom,momd,MaxIdxGare,MaxIdxDspl,MaxIdxCat:Integer;
implementation
uses
    Console1 ;
    {$R *.dfm}
    {$D+}
procedure TForm8.FormCreate(Sender: TObject);
begin
    ClassFlg:=False;
    // Form2.ABSTable3.TableName:='SelezioneGare';
    Form2.ABSTable3.Open;
    Form2.ABSTable3.Edit;
    // Form2.ABSTable3D.TableName:='SelezioneDSPL';
    Form2.ABSTable3D.Open;
    Form2.ABSTable3D.Edit;
    ABSTable15.TableName:='Classifica';
    ABSTable15.Open;
    ABSTable15.Edit;
    ABSTable16R.TableName:='R16R';
    ABSTable16R.Open;
    ABSTable20.TableName:='repDSPL';
    // ABSTable20.Open;
    Form2.ABSTable1.Open;    //registra su Servizio
    Form2.ABSTable1.INSERT;
    Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
    Form2.ABSTable1.FieldName('what').AsString:='Accesso alle Classifiche';
    Form2.ABSTable1.FieldName('OUT').AsString:='';

```

```

Form2.ABSTable1.Post;
Form2.ABSTable1.Close;
Form2.ABSTable6.Open;
Form2.ABSTable8.Open;
if ABSTable15.RecordCount>0 then
    begin

TDateTimeField(Form8.ABSTable15.FieldName('TAME1')).DisplayFormat:='nn:ss,zz
z';

TDateTimeField(Form8.ABSTable15.FieldName('TBME2')).DisplayFormat:='nn:ss,zz
z';

TDateTimeField(Form8.ABSTable15.FieldName('PEN')).DisplayFormat:='nn:ss,zzz'
;
    end;
    ABSTable16.TableName:='Ranking';
    ABSTable18.TableName:='Temp8';
    Form2.ABSTable5.Open;
    Form2.ABSTable4.Open;
    SetLength(ArGare,20);
    SetLength(ArDspl,20);
    SetLength(ArCat,10);
    ABSTable19.TableName:='totale';
    Shape1.Brush.Color := clRed;
    Shape2.Brush.Color := clRed;
    Shape3.Brush.Color := clRed;
    RadioButton1.Color:= clRed;
    RadioButton2.Color:= clRed;
    RadioButton3.Color:= clRed;
end;

procedure TForm8.Button1Click(Sender: TObject); //Seleziona le gare
begin
    DataSource8.DataSet:=Form2.ABSTable5;           //display gare su dbgrid
    DBGrid8.DataSource:=DataSource8;
    DBGrid8.Width:=376;
    DBGrid8.Columns[0].Width:=44;
    DBGrid8.Columns[1].Width:=64;
    DBGrid8.Columns[2].Width:=228;
    SelGare:=True;
    SelDspl:=False;
    SetLength(ArGare,0);
    SetLength(ArGare,20);
    Shape1.Brush.Color := clRed;
end;

procedure TForm8.DBGrid8CellClick(Column: TColumn); //Inserisci-Scegli
begin
    ListBox8.Items.Add(DBGrid8.Fields[0].AsString);
    Self.ActiveControl:=nil;
end;

procedure TForm8.Button2Click(Sender: TObject); //Rimuovere
begin
    if ListBox8.Items.Count > 0 then //verifica che non c'è selezionato
        begin
            if ListBox8.Itemindex =-1 then

```

```

        begin
            ShowMessage('Nessun elemento selezionato');
            Exit;
        end;
    end;
    ListBox8.DeleteSelected;
end;

procedure TForm8.Button3Click(Sender: TObject);    //Salva i valori negli array
var
    k,i:ShortInt;
    kk:Integer;
begin
    for i := ListBox8.Items.Count - 1 downto 0 do    //toglie gli spazi vuoti
        begin
            if (Trim(ListBox8.Items[i]) = '') then
                ListBox8.Items.Delete(i);
            end;
        end;

    if MessageDlg('Confermi gli elementi scelti ?',mtconfirmation,[mbYes,mbNo], 0)
    = mrNo then
        begin
            if SelGare=True then    // verifica la scelta
                begin
                    SetLength(ArGare,0);
                    SetLength(ArGare,20);
                    Shape1.Brush.Color := clRed;
                end;
            if SelDspl=True then
                begin
                    SetLength(ArDspl,0);
                    SetLength(ArDspl,20);
                    Shape2.Brush.Color := clRed;
                end;
            ListBox8.Clear;
            Exit;
        end;

    if SelGare=True then    //Salva le Gare scelte nell'ARRAY
        begin
            for k := 0 to ListBox8.Items.Count-1 do
                begin
                    kk:= StrToInt(ListBox8.Items[k]); // NOTA BENE gli array non
consentono
                    ArGare[k] := kk;    // di eliminare il passaggio tramite
kk

                end;

            for i := 0 to High(ArGare) do //sistema per verificare se l'array è vuoto
            if ArGare[i]=0 then    // va al primo vuoto e segna l'indice precedente
                begin
                    mom := i-1;
                    MaxIdxGare:=mom;
                    Break;
                end;
            if mom=-1 then
                begin
                    ShowMessage('Non vi sono elementi da salvare ! - ripeti');
                end;
            end;
        end;
    end;
end;

```



```

Exit;
end else                                     //reinizializza
begin
    ShowMessage(' Dati salvati'+#13+'Se Classifichi per Gare:
inserisci Gruppo e Categoria'
    +#13#10#10+'Se Classifichi per Discipline, selezionala/e');
    Shapel.Brush.Color := clLime;
end ;

for k := ListBox8.Items.Count - 1 downto 0 do    //resetta la listBox
    ListBox8.Items.Delete(k);
    SelGare:=False;
end;

if SelDspl=True then                         // memorizza le discipline
begin
    for k := 0 to ListBox8.Items.Count-1 do
        begin
            kk:= StrToInt(ListBox8.Items[k]); // NOTA BENE gli array non
consentono
            ArDspl[k] := kk;                  // di eliminare il passaggio tramite
kk
        end;

        for i := 0 to High(ArDspl) do        //Questo è l'unico sistema per
verificare
            if ArDspl[i]=0 then                // se l'array è vuoto
                begin
                    mom := i-1;
                    MaxIdxDspl:=mom;
                    Break;
                end;
            if mom=-1 then
                begin
                    ShowMessage('Non vi sono elementi da salvare ! - ripeti');
                    Exit;
                end else
                    //reinizializza
                begin
                    ShowMessage(' Dati salvati'+#13+'Inserisci la Categoria');
                    Shape2.Brush.Color := clLime;
                end ;
            for k := ListBox8.Items.Count - 1 downto 0 do    //resetta la listBox
                ListBox8.Items.Delete(k);
                SelDspl:=False;
            end;
            DBGrid8.DataSource.DataSet.Close;                //svuota la dbgrid
            DBGrid8.Refresh;
        end;

procedure TForm8.Button4Click(Sender: TObject);        //Seleziona le discipline
var
    k,i:ShortInt;
begin
    for k := ListBox8.Items.Count - 1 downto 0 do    //resetta la listBox
        ListBox8.Items.Delete(k);                    //nel caso fosse già inserita una
gara
        SelGare:=False;
        DataSource8.DataSet:=Form2.ABSTable4;          //display gare su dbgrid

```

```

DBGrid8.DataSource:=DataSource8;
DBGrid8.Width:=376;
DBGrid8.Columns[0].Width:=44;
DBGrid8.Columns[1].Width:=241;
DBGrid8.Columns[2].Width:=50;
Shape2.Brush.Color := clRed;
SelDspl:=True;
SelGare:=False;
SetLength(ArDspl,0);
SetLength(ArDspl,20);
for i := 0 to High(ArGare) do    //Controlla che le gare siano state indicate
    if ArGare[i]=0 then
        begin
            mom := i-1; //mom = all'ultimo elemento dell'array memorizzato
            Break;
        end;
    if mom=-1 then
        begin
            Shape1.Brush.Color := clRed;
            ShowMessage('Scegli prima una o piùgare'+#13+'    e salva');
            DBGrid8.DataSource.DataSet.Close;    //svuota la dbgrid
            DBGrid8.Refresh;
            Exit;
        end;

end;

procedure TForm8.Button6Click(Sender: TObject);    // CLASSIFICARE PER GARE
var
    garesi,dsplsi,trovato:Boolean;
    QryStr:string;
    Cgara,Cdspl,qtrl,i:Integer;
begin
    Form2.ABSTable5.Open;
    Cgara:=0;
    Cdspl:=0;
    Form2.ABSTable3.Close;
    Form2.ABSTable3.EmptyTable; //resetta la tabella gare selezionate
    Form2.ABSTable3.Open;
    for i := 0 to High(ArGare) do
        if ArGare[i]=0 then
            begin
                mom := i-1; //mom = all'ultimo elemento dell'array memorizzato
                Break;
            end;
        if mom=-1 then
            begin
                ShowMessage('Non vi sono elementi memorizzati ! - ripeti');
                Shape1.Brush.Color := clRed;
                Exit;
            end;
    for i:= 0 to mom do    //crea la tabella 3
        begin
            idx:=ArGare[i]; //Trasferisce i dati dall'Array alla Tabella
            Form2.ABSTable3.Insert;
            Form2.ABSTable3.FieldName('Codice').AsInteger:=idx;
            garesi:=Form2.ABSTable5.Locate('CODGA',idx,[]);
            Form2.ABSTable3.FieldName('Descrizione').AsString:=

```

```

        Form2.ABSTable5.FieldName('LUOGO').AsString;
        Form2.ABSTable3.Post;
    end;

    if comboBox1.ItemIndex=-1 then      //verifica Categoria
    begin
        ShowMessage('Devi scegliere una Categoria');
        Exit;
    end else
    begin
        if MessageDlg('Confermi di effettuare la classifica per la Categoria
'+comboBox1.Text+' ?',
            mtconfirmation, [mbYes,mbNo], 0) = mrNo then
            begin
                Shape3.Brush.Color := clRed;
                Exit ;
            end else
                shape3.Brush.Color:=clLime;
            end;
        //Verifica esistenza Gruppo
        if (RadioButton2.Checked=False) and (RadioButton1.Checked=False) and
(RadioButton3.Checked=False ) then
            begin
                ShowMessage('Devi scegliere un GRUPPO');
                Exit;
            end;

            //DataSource8 è associato alla tabella Gare Selezionate
            DataSource8.DataSet:=Form2.ABSTable3;      //display gare su dbgrid
            DBGrid8.DataSource:=DataSource8;
            DBGrid8.Width:=376;
            DBGrid8.Columns[0].Width:=44;
            DBGrid8.Columns[1].Width:=291;
            if MessageDlg('Confermi di effettuare la classifica come selezionato?',
                mtconfirmation, [mbYes,mbNo], 0) = mrNo then
                begin
                    Form2.ABSTable3.Close;
                    Form2.ABSTable3.EmptyTable;
                    Shape1.Brush.Color := clRed;
                    SetLength(ArGare,0); //azzera l'array delle Gare non confermando la scelta
                    SetLength(ArGare,20);
                    Exit;
                end;

                //GRUPPO BASE
            if RadioButton1.Checked=True then
                if MessageDlg('Confermi di effettuare la classifica sul Gruppo Base ?',
                    mtconfirmation, [mbYes,mbNo], 0) = mrNo then
                    begin
                        ShowMessage('Cambia il Gruppo');
                        Exit;
                    end else
                    begin //converte l'array delle gare in string per il Query
                        if Form2.ABSTable6.RecordCount=0 then
                            begin
                                ShowMessage('Attenzione il Gruppo scelto non contiene discipline
!');
                                Exit;
                            end;
                    end;
                end;
            end;
        end;
    end;

```

```

        QryStr:='(';
        for i := 0 to MaxIdxGare-1 do
            QryStr:=QryStr+IntToStr(ArGare[i])+',';
            QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
            QryStr:=QryStr+')';

        with ABSQuery80 do           //Estrae dal Base le discipline gruppo Base
            begin                   //per tutte le gare indicate e estrae le
                Close;              // coppie di valori n.gara e n.disciplina
                SQL.Text:='Select * from Base where CODGA IN '+QryStr;

//dall'ARRAY
                ExecSQL;
                Open;
            end;
            if ABSQuery80.RecordCount=0 then
                begin
                    ShowMessage('Attenzione il Gruppo scelto non contiene discipline
!');

                    Exit;
                end;
                //Fase di raccolta record da classificare in T15 Classifica
            if ABSTable15.RecordCount>0 then //Azzero la tabella Classifica
                begin                 // preesistente per non causare
                    ABSTable15.Close; // problemi di indice unique
                    ABSTable15.EmptyTable;
                    ABSTable15.Open;
                end;
                ABSTable15.First; //era "last" ?
                ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
                qtrl:=ABSQuery80.RecordCount;
                while not ABSQuery80.EOF do // selezionate
                    begin
                        Cgara:=ABSQuery80.FieldName('CODGA').AsInteger;
                        Cdspl:=ABSQuery80.FieldName('CODDSPL').AsInteger;
                        with ABSQuery85 do //Estrae dalle Gare le discipline da
classif
discipline
                            begin //Costruisce Classifica con tutti i record gare e
                                Close; //selezionate e gli Atleti che le hanno in comune
                                SQL.Text:='INSERT INTO Classifica SELECT * From Gara where
CODGA='
                                    +QuotedStr(IntToStr(Cgara))+ ' AND
CODDSPL='+QuotedStr(IntToStr(Cdspl))
                                    + 'AND CAT =' + quotedstr(comboBox1.text);
                                ExecSQL;
                            end;
                                qtrl:=qtrl-1;
                                ABSQuery80.Next;
                            end;
                                ComboBox1.ItemIndex:=-1;
                                if qtrl>0 then ShowMessage('ABSQuery80 errore eof');

                                with ABSQuery85 do
                                    begin
                                        Close;
                                        SQL.Text:='Select * from Classifica Order By
"COGNOME","CODDSPL"';
                                        ExecSQL;

```

```

        Open;
    end;                                //ricostruisce la tabella 15 dopo aver ordinato i
dati
ABSTable15.Close;    //devo cancellare per poter sostituire
ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
ABSTable15.Open;
ABSTable15.Last;
ABSQuery85.First;
qtrl:=ABSQuery85.RecordCount;
15 while not ABSQuery85.EOF do //copia il query ordinato nella Tabella

    begin
        ABSTable15.Insert;
        ABSTable15.FieldName('NMR').AsInteger:=
            ABSQuery85.FieldName('NMR').AsInteger;
        ABSTable15.FieldName('CODGA').AsInteger:=
            ABSQuery85.FieldName('CODGA').AsInteger;
        ABSTable15.FieldName('CODDSPL').AsInteger:=
            ABSQuery85.FieldName('CODDSPL').AsInteger;
        ABSTable15.FieldName('CODATL').AsInteger:=
            ABSQuery85.FieldName('CODATL').AsInteger;
        ABSTable15.FieldName('COGNOME').AsString:=
            ABSQuery85.FieldName('COGNOME').AsString;
        ABSTable15.FieldName('NMPET').AsInteger:=
            ABSQuery85.FieldName('NMPET').AsInteger;
        ABSTable15.FieldName('NMPOS').AsInteger:=
            ABSQuery85.FieldName('NMPOS').AsInteger;
        ABSTable15.FieldName('TAME1').AsDateTime:=
            ABSQuery85.FieldName('TAME1').AsDateTime;
        ABSTable15.FieldName('TBME2').AsDateTime:=
            ABSQuery85.FieldName('TBME2').AsDateTime;
        ABSTable15.FieldName('DQ').AsBoolean:=
            ABSQuery85.FieldName('DQ').AsBoolean;
        ABSTable15.FieldName('PEN').AsDateTime:=
            ABSQuery85.FieldName('PEN').AsDateTime;
        ABSTable15.FieldName('MANL').AsBoolean:=
            ABSQuery85.FieldName('MANL').AsBoolean;
        ABSTable15.FieldName('CAT').AsInteger:=
            ABSQuery85.FieldName('CAT').AsInteger;
        qtrl:=qtrl-1;
        ABSTable15.Post;
        ABSQuery85.Next;
    end;
    if qtrl>0 then ShowMessage('ABSQuery85 errore eof');

    if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la
Classifica ?'
,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
    begin
        frxReport8.ShowReport;    //dati da ABSTable15 (Classifica)
        if MessageDlg('Confermi i dati a base della classifica? '+#13+
'In caso negativo ESCI e correggi le scelte fatte.',
mtconfirmation, [mbYes,mbNo], 0) = mrNO then
        begin
            ABSTable15.Close;
            ABSTable15.EmptyTable;
            Exit;
        end;
    end;
end;

```

```

        end;

        RadioButton1.Color:= clLime;
        Calcola(Self);
        Exit;
    end;

    //GRUPPO A
    if RadioButton2.Checked=True then
        if MessageDlg('Confermi di effettuare la classifica sul Gruppo A ?',
            mtconfirmation, [mbYes,mbNo], 0) = mrNo then
            begin
                ShowMessage('Cambia il Gruppo');
                Exit;
            end else
            begin //converte l'array delle gare in string per il Query
                if Form2.ABSTable8.RecordCount=0 then
                    begin
                        ShowMessage('Attenzione il Gruppo scelto non contiene discipline
!');

                        Exit;
                    end;
                    QryStr:='(';
                    for i := 0 to MaxIdxGare-1 do
                        QryStr:=QryStr+IntToStr(ArGare[i])+',';
                        QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
                        QryStr:=QryStr+')';

                    with ABSQuery80 do //Estrae dal Base le discipline gruppo A
                        begin //per tutte le gare indicate e estrae le
                            Close; // coppie di valori n.gara e n.disciplina
                            SQL.Text:='Select * from GRPA where CODGA IN '+QryStr;
//dall'ARRAY
                            ExecSQL;
                            Open;
                        end; //Fase di raccolta record da classificare in T15 Classifica
                    if ABSQuery80.RecordCount=0 then
                        begin
                            ShowMessage('Attenzione il Gruppo scelto non contiene discipline
!');

                            Exit;
                        end;

                    if ABSTable15.RecordCount>0 then //Azzerò la tabella Classifica
                        begin // preesistente per non causare
                            ABSTable15.Close; // problemi di indice unique
                            ABSTable15.EmptyTable;
                            ABSTable15.Open;
                        end;
                    ABSTable15.Last;
                    ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
                    qtrl:=ABSQuery80.RecordCount;
                    while not ABSQuery80.EOF do // selezionate
                        begin
                            Cgara:=ABSQuery80.FieldName('CODGA').AsInteger;
                            Cdspl:=ABSQuery80.FieldName('CODDSPL').AsInteger;
                            with ABSQuery85 do //Estrae dalle Gare le discipline da
                                classif

```

```

discipline      begin    //Costruisce Classifica con tutti i record gare e

                  Close; //selezionate e gli Atleti che le hanno in comune
                  SQL.Text:='INSERT INTO Classifica SELECT * From Gara where
CODGA='
                  +QuotedStr(IntToStr(Cgara))+ ' AND
CODDSPL='+QuotedStr(IntToStr(Cdspl))
                  + 'AND CAT =' + quotedstr(comboBox1.text);
                  ExecSQL;
                  end;
                  qtrl:=qtrl-1;
                  ABSQuery80.Next;
                  end;
                  ComboBox1.ItemIndex:=-1;
                  if qtrl>0 then ShowMessage('ABSQuery80 errore eof');

                  with ABSQuery85 do
                  begin
                  Close;
                  SQL.Text:='Select * from Classifica Order By
"COGNOME","CODDSPL"';
                  ExecSQL;
                  Open;
                  end;
                  ABSTable15.Close;    //30/04/15 18:06 devo cancellare per poter
sostituire
                  ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
                  ABSTable15.Open;
                  ABSTable15.Last;
                  ABSQuery85.First;
                  qtrl:=ABSQuery85.RecordCount;
                  while not ABSQuery85.EOF do //copia il query ordinato nella Tabella
15
                  begin
                  ABSTable15.Insert;
                  ABSTable15.FieldName('NMR').AsInteger:=
                      ABSQuery85.FieldName('NMR').AsInteger;
                  ABSTable15.FieldName('CODGA').AsInteger:=
                      ABSQuery85.FieldName('CODGA').AsInteger;
                  ABSTable15.FieldName('CODDSPL').AsInteger:=
                      ABSQuery85.FieldName('CODDSPL').AsInteger;
                  ABSTable15.FieldName('CODATL').AsInteger:=
                      ABSQuery85.FieldName('CODATL').AsInteger;
                  ABSTable15.FieldName('COGNOME').AsString:=
                      ABSQuery85.FieldName('COGNOME').AsString;
                  ABSTable15.FieldName('NMPET').AsInteger:=
                      ABSQuery85.FieldName('NMPET').AsInteger;
                  ABSTable15.FieldName('NMPOS').AsInteger:=
                      ABSQuery85.FieldName('NMPOS').AsInteger;
                  ABSTable15.FieldName('TAME1').AsDateTime:=
                      ABSQuery85.FieldName('TAME1').AsDateTime;
                  ABSTable15.FieldName('TBME2').AsDateTime:=
                      ABSQuery85.FieldName('TBME2').AsDateTime;
                  ABSTable15.FieldName('DQ').AsBoolean:=
                      ABSQuery85.FieldName('DQ').AsBoolean;
                  ABSTable15.FieldName('PEN').AsDateTime:=
                      ABSQuery85.FieldName('PEN').AsDateTime;
                  ABSTable15.FieldName('MANL').AsBoolean:=

```

```

        ABSQuery85.FieldName('MANL').AsBoolean;
        ABSTable15.FieldName('CAT').AsInteger:=
            ABSQuery85.FieldName('CAT').AsInteger;
        qtrl:=qtrl-1;
        ABSTable15.Post;
        ABSQuery85.Next;
    end;
    if qtrl>0 then ShowMessage('ABSQuery85 errore eof');

    if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la
Classifica ?'
,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
    begin
        frxReport8.ShowReport;
        if MessageDlg('Confermi i dati a base della classifica? '+#13+
'In caso negativo ESCI e correggi le scelte fatte.',
mtconfirmation, [mbYes,mbNo], 0) = mrNO then
        begin
            ABSTable15.Close;
            ABSTable15.EmptyTable;
            Exit;
        end;
    end;

    end;

    RadioButton2.Color:= clLime;
    Calcola(Self);
    Exit;
end;

//GRUPPO B
if RadioButton3.Checked=True then
    if MessageDlg('Confermi di effettuare la classifica sul Gruppo B ?',
mtconfirmation, [mbYes,mbNo], 0) = mrNo then
    begin
        ShowMessage('Cambia il Gruppo');
        Exit;
    end else
    begin //converte l'array delle gare in string per il Query
        Form2.ABSTable9.Open;
        if Form2.ABSTable9.RecordCount=0 then
            begin
                ShowMessage('Attenzione il Gruppo scelto non contiene discipline
!');

                Exit;
            end;
            QryStr:='(';
            for i := 0 to MaxIdxGare-1 do
                QryStr:=QryStr+IntToStr(ArGare[i])+',';
                QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
                QryStr:=QryStr+')';

            with ABSQuery80 do //Estrae dal Base le discipline gruppo B
            begin //per tutte le gare indicate e estrae le
                Close; // coppie di valori n.gara e n.disciplina
                SQL.Text:='Select * from GRPB where CODGA IN '+QryStr;

//dall'ARRAY
                ExecSQL;

```



```

        Open;
    end; //Fase di raccolta record da classificare in T15 Classifica
    if ABSQuery80.RecordCount=0 then
        begin
            ShowMessage ('Attenzione il Gruppo scelto non contiene discipline
!');

            Exit;
        end;

    if ABSTable15.RecordCount>0 then //Azzero la tabella Classifica
        begin // preesistente per non causare
            ABSTable15.Close; // problemi di indice unique
            ABSTable15.EmptyTable;
            ABSTable15.Open;
        end;
    ABSTable15.Last;
    ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
    qtrl:=ABSQuery80.RecordCount;
    while not ABSQuery80.EOF do // selezionate
        begin
            Cgara:=ABSQuery80.FieldName('CODGA').AsInteger;
            Cdspl:=ABSQuery80.FieldName('CODDSPL').AsInteger;
            with ABSQuery85 do //Estrae dalle Gare le discipline da
                begin //Costruisce Classifica con tutti i record gare e
                    Close; //selezionate e gli Atleti che le hanno in comune
                    SQL.Text:='INSERT INTO Classifica SELECT * From Gara where
CODGA='
                    +QuotedStr(IntToStr(Cgara))+ ' AND
CODDSPL='+QuotedStr(IntToStr(Cdspl))
                    + 'AND CAT =' + quotedstr(comboBox1.text);
                    ExecSQL;
                end;
                qtrl:=qtrl-1;
                ABSQuery80.Next;
            end;
        ComboBox1.ItemIndex:=-1;
        if qtrl>0 then ShowMessage ('ABSQuery80 errore eof');

            with ABSQuery85 do //per ordinare per cognome la classifica
                begin
                    Close;
                    SQL.Text:='Select * from Classifica Order By
"COGNOME","CODDSPL"';
                    ExecSQL;
                    Open;
                end;
            ABSTable15.Close; //30/04/15 18:06 devo cancellare per poter
            sostituire
            ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
            ABSTable15.Open;
            ABSTable15.Last;
            ABSQuery85.First;
            qtrl:=ABSQuery85.RecordCount;
            while not ABSQuery85.EOF do //copia il query ordinato per
                cogn/discipl nella Tabella 15
                begin

```

```

ABSTable15.Insert;
ABSTable15.FieldName('NMR').AsInteger:=
    ABSQuery85.FieldName('NMR').AsInteger;
ABSTable15.FieldName('CODGA').AsInteger:=
    ABSQuery85.FieldName('CODGA').AsInteger;
ABSTable15.FieldName('CODDSPL').AsInteger:=
    ABSQuery85.FieldName('CODDSPL').AsInteger;
ABSTable15.FieldName('CODATL').AsInteger:=
    ABSQuery85.FieldName('CODATL').AsInteger;
ABSTable15.FieldName('COGNOME').AsString:=
    ABSQuery85.FieldName('COGNOME').AsString;
ABSTable15.FieldName('NMPET').AsInteger:=
    ABSQuery85.FieldName('NMPET').AsInteger;
ABSTable15.FieldName('NMPOS').AsInteger:=
    ABSQuery85.FieldName('NMPOS').AsInteger;
ABSTable15.FieldName('TAME1').AsDateTime:=
    ABSQuery85.FieldName('TAME1').AsDateTime;
ABSTable15.FieldName('TBME2').AsDateTime:=
    ABSQuery85.FieldName('TBME2').AsDateTime;
ABSTable15.FieldName('DQ').AsBoolean:=
    ABSQuery85.FieldName('DQ').AsBoolean;
ABSTable15.FieldName('PEN').AsDateTime:=
    ABSQuery85.FieldName('PEN').AsDateTime;
ABSTable15.FieldName('MANL').AsBoolean:=
    ABSQuery85.FieldName('MANL').AsBoolean;
ABSTable15.FieldName('CAT').AsInteger:=
    ABSQuery85.FieldName('CAT').AsInteger;
qtrl:=qtrl-1;
ABSTable15.Post;
ABSQuery85.Next;
end;
if qtrl>0 then ShowMessage('ABSQuery85 errore eof');

if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la
Classifica ?',
,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
begin
    frxReport8.ShowReport;
    if MessageDlg('Confermi i dati a base della classifica? '+#13+
    'In caso negativo ESCI e correggi le scelte fatte.',
    mtconfirmation, [mbYes,mbNo], 0) = mrNO then
    begin
        ABSTable15.Close;
        ABSTable15.EmptyTable;
        Exit;
    end;
end;

RadioButton3.Color:= clLime;
Calcola(Self);
Exit;
end;
end;

procedure TForm8.Button7Click(Sender: TObject); //Classifica per discipline
var
    dsplsi,garesi,trovato:Boolean;           //Individua prima le Gare
    qtrl,qtrl2,cgara,cdspl,i:Integer;

```

```

begin
    Form2.ABSTable5.Open;
    Form2.ABSTable4.Open;
    ClassDp:=True;
    cgara:=0;
    cdspl:=0;
    Form2.ABSTable3.Close;
    Form2.ABSTable3.EmptyTable; //resetta la tabella gare selezionate
    Form2.ABSTable3.Open; // e carica la Tabella con le Gare scelte per
    for i := 0 to High(ArGare) do //la ricerca per discipline
        if ArGare[i]=0 then
            begin
                mom := i-1; //mom = all'ultimo elemento dell'array memorizzato
                Break;
            end;
        if mom=-1 then
            begin
                ShowMessage('Non vi sono elementi memorizzati ! - ripeti');
                Shape2.Brush.Color := clRed;
                Exit;
            end;
        for i:= 0 to mom do //crea la tabella 3
            begin
                idx:=ArGare[i]; //Trasferisce i dati dall'Array alla Tabella
                Form2.ABSTable3.Insert;
                Form2.ABSTable3.FieldName('Codice').AsInteger:=idx;
                garesi:=Form2.ABSTable5.Locate('CODGA',idx,[]);
                Form2.ABSTable3.FieldName('Descrizione').AsString:=
                    Form2.ABSTable5.FieldName('LUOGO').AsString;
                Form2.ABSTable3.Post;
            end;

    Form2.ABSTable3D.Close;
    Form2.ABSTable3D.EmptyTable; //resetta la tabella discipline selezionate
    Form2.ABSTable3D.Open;
    for i := 0 to High(ArDspl) do
        if ArDspl[i]=0 then
            begin
                momd := i-1; //momd = all'ultimo elemento dell'array memorizzato
                Break;
            end;
        if momd=-1 then
            begin
                DBGrid8.DataSource.DataSet.Close;
                ShowMessage('Non vi sono discipline memorizzate ! - Selezionale'
                    +#13#10+' In alternativa Esci ');
                Exit;
            end;
        for i:= 0 to momd do //crea la tabella 3D
            begin
                idx:=ArDspl[i]; //Trasferisce i dati dall'Array alla Tabella 3D
                Form2.ABSTable3D.Insert;
                Form2.ABSTable3D.FieldName('CODDSPL').AsInteger:=idx;
                dsplsi:=Form2.ABSTable4.Locate('CODDSPL',idx,[]);
                Form2.ABSTable3D.FieldName('DESCR').AsString:=
                    Form2.ABSTable4.FieldName('DESCR').AsString;
                Form2.ABSTable3D.FieldName('TMAX').AsDateTime:=

```

```

        Form2.ABSTable4.FieldName('TMAX').AsDateTime;
    Form2.ABSTable3D.Post;
end;                                     //mi sembra una conferma inutile
                                     //Datasource8 è ora associato alle discipline scelte
{   DataSource8.DataSet:=Form2.ABSTable3D;           //display gare su dbgrid
    DBGrid8.DataSource:=DataSource8;
    DBGrid8.Width:=376;
    DBGrid8.Columns[0].Width:=44;
    DBGrid8.Columns[1].Width:=291;
    if MessageDlg('Confermi di effettuare la classifica su questi elementi ?',
        mtconfirmation, [mbYes,mbNo], 0) = mrNo then
        begin
            Form2.ABSTable3D.Close;
            Form2.ABSTable3D.EmptyTable;
            Shape1.Brush.Color := clRed;
            SetLength(ArDspl,0); //azzerare l'array delle Discipline non confermando la
scelta
            SetLength(ArDspl,20);
            Exit;
        end; }
    if comboBox1.ItemIndex=-1 then
        begin
            ShowMessage('Devi scegliere una Categoria');
            Exit;
        end else
        begin
            if MessageDlg('Confermi di effettuare la classifica per la Categoria
'+comboBox1.Text+' ?',
                mtconfirmation, [mbYes,mbNo], 0) = mrNo then
                begin
                    Shape3.Brush.Color := clRed;
                    Exit ;
                end ;
            end;                                     //inizia a raccogliere i record da classificare
//lavori in corso
            Shape3.Brush.Color := clLime;
            if ABSTable15.RecordCount>0 then          //Azzero la tabella Classifica
                begin                                // preesistente per non causare
                    ABSTable15.Close;                // problemi di indice unique
                    ABSTable15.EmptyTable;
                    ABSTable15.Open;
                end;
                ABSTable15.Last;
                Form2.ABSTable3.First;                //Form2.ABSTable3 è una tabella con gare
                qtrl:=Form2.ABSTable3.RecordCount;
while not Form2.ABSTable3.EOF do                    //per ogni gara selezionata
    begin
        Cgara:=Form2.ABSTable3.FieldName('Codice').AsInteger;
        qtrl2:=Form2.ABSTable3D.RecordCount;
        Form2.ABSTable3D.First;
        while not Form2.ABSTable3D.EOF do           //per ogni disciplina selezionata
            begin
                Cdspl:=Form2.ABSTable3D.FieldName('CODDSPL').AsInteger;
                with ABSQuery85 do                   //Estrae dalle Gare le discipline da
classif
                    begin                             //Costruisce Classifica con tutti i record gare e
discipline
                        Close; //selezionate e gli Atleti che le hanno in comune

```

```

SQL.Text:='INSERT INTO Classifica SELECT * From Gara where
CODGA='
+QuotedStr(IntToStr(Cgara))+ ' AND
CODDSPL='+QuotedStr(IntToStr(Cdspl))
+ 'AND CAT =' + quotedstr(comboBox1.text);
ExecSQL;
end;

qtrl2:=qtrl2-1;
Form2.ABSTable3D.Next
end;
if qtrl2>0 then ShowMessage('Elenco discipline errore eof');
qtrl:=qtrl-1;
Form2.ABSTable3.Next;
end;
ComboBox1.ItemIndex:=-1;
if qtrl>0 then ShowMessage('Elenco gare errore eof');

with ABSQuery85 do
begin
Close;
SQL.Text:='Select * from Classifica Order By
"COGNOME","CODDSPL"';
ExecSQL;
Open;
end;
ABSTable15.Close; //30/04/15 18:06 devo cancellare per poter
sostituire
ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
ABSTable15.Open;
ABSTable15.Last;
ABSQuery85.First;
qtrl:=ABSQuery85.RecordCount;
while not ABSQuery85.EOF do //copia il query ordinato nella Tabella
15
begin
ABSTable15.Insert;
ABSTable15.FieldName('NMR').AsInteger:=
ABSQuery85.FieldName('NMR').AsInteger;
ABSTable15.FieldName('CODGA').AsInteger:=
ABSQuery85.FieldName('CODGA').AsInteger;
ABSTable15.FieldName('CODDSPL').AsInteger:=
ABSQuery85.FieldName('CODDSPL').AsInteger;
ABSTable15.FieldName('CODATL').AsInteger:=
ABSQuery85.FieldName('CODATL').AsInteger;
ABSTable15.FieldName('COGNOME').AsString:=
ABSQuery85.FieldName('COGNOME').AsString;
ABSTable15.FieldName('NMPET').AsInteger:=
ABSQuery85.FieldName('NMPET').AsInteger;
ABSTable15.FieldName('NMPOS').AsInteger:=
ABSQuery85.FieldName('NMPOS').AsInteger;
ABSTable15.FieldName('TAME1').AsDateTime:=
ABSQuery85.FieldName('TAME1').AsDateTime;
ABSTable15.FieldName('TBME2').AsDateTime:=
ABSQuery85.FieldName('TBME2').AsDateTime;
ABSTable15.FieldName('DQ').AsBoolean:=
ABSQuery85.FieldName('DQ').AsBoolean;
ABSTable15.FieldName('PEN').AsDateTime:=

```

```

        ABSQuery85.FieldName('PEN').AsDateTime;
ABSTable15.FieldName('MANL').AsBoolean:=
        ABSQuery85.FieldName('MANL').AsBoolean;
ABSTable15.FieldName('CAT').AsInteger:=
        ABSQuery85.FieldName('CAT').AsInteger;
qtrl:=qtrl-1;
ABSTable15.Post;
ABSQuery85.Next;
    end;
    if qtrl>0 then ShowMessage('ABSQuery85 errore eof');

    if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la
Classifica ?'
,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
    begin
        frxReport8.ShowReport;
        if MessageDlg('Confermi i dati a base della classifica? '+#13+
        'In caso negativo ESCI e correggi le scelte fatte.',
        mtconfirmation, [mbYes,mbNo], 0) = mrNO then
        begin
            ABSTable15.Close;
            ABSTable15.EmptyTable;
            Exit;
        end;
    end;

    Calcola(Self);
    Exit;
end;

procedure TForm8.Calcola(Sender: TObject);    // C A L C O L A
var                                            //in questa fase i dati di base
t1,t2,p1,tmx,tmcn,d1,d2:TDateTime;          //confermati e sono su T15
h1,m1,s1,z1,h2,m2,s2,z2,h3,m3,s3,z3:Word;
sql,esclusioni,trovato,ATLsi,DQSi:Boolean;
esclusi:TStringList;
squalificati:TStringList;
ca,dp,ga,qtrl,ct,i,imax,temp2,tt,tt2,tln,t2n:Integer;
{dacancA:array of smallInt;
dacancD:array of smallInt;
dacancG:array of smallInt; }
cgn,dscrz,durata:string;
RnkMax,rn,CODATL16:Integer;                  //Fase segnalazione atleti in difetto
begin
    formatDateTime('nn:ss,zzz',tmx);
    formatDateTime('nn:ss,zzz',tmcn);
    formatDateTime('nn:ss,zzz',t1);
    formatDateTime('nn:ss,zzz',t2);
    formatDateTime('nn:ss,zzz',p1);
    trovato:=False;
    DQSi:=False;    //serve per assegnare il punteggio 0
{ SetLength(dacancA,0); //resetta l'array che conterrà codici atleti
SetLength(dacancA,300);
SetLength(dacancD,0); //resetta l'array che conterrà codici discipline
SetLength(dacancD,300);
SetLength(dacancG,0); //resetta l'array che conterrà codici gare
SetLength(dacancG,300); }
    esclusi:=TStringList.Create;//NOTA BENE L'ESCLUSIONE VALE SOLO NEL RUN IN CORSO

```

```

squalificati:= TStringList.Create;
ABSTable16.Close;
ABSTable16.EmptyTable; //resetta la tabella che conterrà dati classificati con
il rank
ABSTable15.Open; //tabella dati da classificare
ABSTable15.Edit;
ABSTable15.Last; //per ottenere un conteggio corretto ???!!!
ABSTable15.First;
qtrl:=ABSTable15.RecordCount; //conta i dati in classifiche
//i:=1;
esclusioni:=False;

//analizza i dati di classifica raccolti in T15 ed evidenzia i problemi
while not ABSTable15.Eof do //memorizzazione e correzione
begin
    t1:=TimeOf(ABSTable15.FieldName('TAME1').AsDateTime);
    t2:=TimeOf(ABSTable15.FieldName('TBME2').AsDateTime);
    DecodeTime(t1,h1,m1,s1,z1);
        t1n:=h1+m1+s1+z1;
    DecodeTime(t2,h2,m2,s2,z2);
        t2n:=h2+m2+s2+z2;
    sql:=ABSTable15.FieldName('DQ').AsBoolean;
    ct:=ABSTable15.FieldName('CAT').AsInteger;
    ca:= ABSTable15.FieldName('CODATL').AsInteger; //memorizza i record
    cgn:=ABSTable15.FieldName('COGNOME').AsString;
    dp:= ABSTable15.FieldName('CODDSPL').AsInteger;
    ga:=ABSTable15.FieldName('CODGA').AsInteger;
    if ct=0 then //manca la categoria e quindi fa uscire dalla procedura
        begin
            ShowMessage('Attenzione l''Atleta '+cgn+' codice '+IntToStr(ca)+'
gara '
                +IntToStr(ga)+' disciplina '+IntToStr(dp)+#13+#13+
                ' non ha la categoria. Correggi');
            Exit;
        end;
    if ((t1n=0) and (t2n=0)) then //Individue mancanza di tempi o squalifica
        begin //salva nella lista esclusi e poi in un file txt
            esclusi.Add('Gara '+Inttostr(ga)+' codice: '+intTostr(ca)+' '+cgn+
                ' Disc. '+intTostr(dp));
            { dacancA[i]:=ca; //memorizza i dati da-completare;
              dacancD[i]:=dp; //Questi array non vengono utilizzati
              dacancG[i]:=ga;
              i:=i+1; }
            esclusioni:=True; //alza un flag per esistenza mancanza dati
        end;

        if sql=True then //Individue una squalifica
            //salva nella lista squalificati e poi in un file txt
            squalificati.Add('Gara '+Inttostr(ga)+' codice: '+intTostr(ca)+'
'+cgn+
                ' Disc. '+intTostr(dp));
            qtrl:=qtrl-1;
            //imax:=i;

            ABSTable15.Next;
        end;
    if qtrl>0 then ShowMessage('ABSTable15 errore eof');

```

```

if (esclusi.Count<50) and (esclusi.Count>0) then
    ShowMessage(' Atleti con dati mancanti' +#13+#13+ esclusi.Text+#13+#13+
    ' Dati salvati su Mancanti.txt') else
        begin
            if esclusi.count>0 then
                ShowMessage('Lista Atleti con dati mancanti, salvata in
"Mancanti.txt".');
            end;
            esclusi.SaveToFile('.\Mancanti.txt');
            esclusi.Clear;    //resetta la lista esclusi
if (squalificati.Count<50) and (squalificati.Count>0) then
            ShowMessage(' Atleti con squalifica'+#13#10+ squalificati.text);
            squalificati.Clear;
                //Calcolo dei tempi  usa SelezioneGare T3 e classifiche T15
Form2.ABSTable3.Open; //è la tabella dove finiscono le gare selezionate
Form2.ABSTable3.First;
while not Form2.ABSTable3.eof do    //Quì inizia il giro delle gare scelte
    begin
        ga:=Form2.ABSTable3.FieldName('Codice').AsInteger;
        ABSTable18.Close;    //azzerla la T18 (discipline della gara)
        ABSTable18.EmptyTable; // per ricevere i dati della nuova gara
        // DataSourceQry.DataSet:=ABSQuery85;
        with ABSQuery85 do //seleziona tutte le discipline della gara scelta
            begin
                Close;    //quì nasce la ABSTable18 con tutte le discipline di una gara
                SQL.Text:='INSERT INTO Temp8 Select * from Classifica where CODGA = '
                + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
                ExecSQL;
                //Open;
            end;    //creata la Temp8 = le discipline della gara indipendentemente
dal Gruppo
                //Crea le discipline da classificare per la gara scelta

if ClassDp=False then //se non effettua una classifica per discipline
    Begin    // cioè sta facendo una classifica per Gare
        with ABSQuery80 do
            begin
                Close;    //determina il Gruppo scelto
                if RadioButton1.Checked=True then
                    SQL.Text:= 'Select CODDSPL from Base where CODGA = '
                    + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
                if RadioButton2.Checked=True then
                    SQL.Text:= 'Select CODDSPL from GRPA where CODGA = '
                    + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
                if RadioButton3.Checked=True then
                    SQL.Text:= 'Select CODDSPL from GRPB where CODGA = '
                    + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
                ExecSQL;
                Open;
            end;
        end else
        begin
            with ABSQuery80 do //Contiene le discipline del Gruppo scelto
            begin
                Close; //seleziona i valori diversi = distinct
                SQL.Text:= 'Select distinct CODDSPL from Classifica where CODGA = '
                + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
            end
        end
    end

```



```

        ExecSQL;
        Open;
    end;
end;

ABSTable18.Open; //contiene le discipline della gara ed i dati relativi

TDateTimeField(Form8.ABSTable18.FieldName('TAME1')).DisplayFormat:='nn:ss,zz
z';

TDateTimeField(Form8.ABSTable18.FieldName('TBME2')).DisplayFormat:='nn:ss,zz
z';

TDateTimeField(Form8.ABSTable18.FieldName('PEN')).DisplayFormat:='nn:ss,zzz'
;

// T18 tutte le dspl di una gara

ABSQuery80.First; // Tutte le dspl del gruppo scelto
while not ABSQuery80.eof do
begin
    dp:=ABSQuery80.FieldName('CODDSPL').AsInteger;
    with ABSQuery83 do //Tutte le discipline
    begin
        Close; //la ABSTable18 fornisce i dati per la disciplina
        SQL.Text:='Select * from Temp8 where CODDSPL = '+
QuotedStr(IntToStr(dp))+
        ' Order By "CODATL"';
        ExecSQL;
        Open;
    end; //Creata Query83 = i record di una disciplina per volta
    //quìci sono tutti i record relativi ad una gara su una disciplina
    //con i tempi di ciascun Atleta e si può valutare il vincitore
    //ATTENZIONE PUO' CAPITARE CHE NESSUN ATLETA È ISCRITTO
    if ABSQuery83.RecordCount=0 then
    begin
        ShowMessage('Attenzione nessun atleta iscritto alla disciplina
'+IntToStr(dp)+'#13+
        'nella Gara : '+IntToStr(ga));
        ABSQuery80.Next;
        Continue;
    end;

ABSTable16.Close;
ABSTable16.EmptyTable; //ripulisce la T16 che conterrà i dati di
ranking
ABSTable16.Open; //definisce i formati dei tempi
TDateTimeField(Form8.ABSTable16.FieldName('TAME1')).
    DisplayFormat:='nn:ss,zzz';
TDateTimeField(Form8.ABSTable16.FieldName('TBME2')).
    DisplayFormat:='nn:ss,zzz';
TDateTimeField(Form8.ABSTable16.FieldName('PEN')).
    DisplayFormat:='nn:ss,zzz';
TDateTimeField(Form8.ABSTable16.FieldName('TMC')).
    DisplayFormat:='nn:ss,zzz';
ABSQuery83.Last; //CREA LA T16 DOVE SI ASSEGNANO I TEMPI e i DQ
ABSQuery83.First; //ricorda che Q83 contiene i dati una
disciplina per volta

```

```

qtrl:=ABSQuery83.RecordCount;
while not ABSQuery83.EOF do //query ordinato nella Tabella 16 Una
Gara
    begin
        ABSTable16.Insert;
        ABSTable16.FieldName('NMR').AsInteger:=
disciplina
            ABSQuery83.FieldName('NMR').AsInteger; // Una

        ABSTable16.FieldName('CODGA').AsInteger:=
            ABSQuery83.FieldName('CODGA').AsInteger;
        ABSTable16.FieldName('CODDSPL').AsInteger:=
            ABSQuery83.FieldName('CODDSPL').AsInteger;
        ABSTable16.FieldName('CODATL').AsInteger:=
            ABSQuery83.FieldName('CODATL').AsInteger;
        ABSTable16.FieldName('COGNOME').AsString:=
            ABSQuery83.FieldName('COGNOME').AsString;
        ABSTable16.FieldName('NMPET').AsInteger:=
            ABSQuery83.FieldName('NMPET').AsInteger;
        ABSTable16.FieldName('NMPOS').AsInteger:=
            ABSQuery83.FieldName('NMPOS').AsInteger;
        ABSTable16.FieldName('TAME1').AsDateTime:=
            ABSQuery83.FieldName('TAME1').AsDateTime;
        ABSTable16.FieldName('TBME2').AsDateTime:=
            ABSQuery83.FieldName('TBME2').AsDateTime;
        ABSTable16.FieldName('DQ').AsBoolean:=
            ABSQuery83.FieldName('DQ').AsBoolean;
        ABSTable16.FieldName('PEN').AsDateTime:=
            ABSQuery83.FieldName('PEN').AsDateTime;
        ABSTable16.FieldName('MANL').AsBoolean:=
            ABSQuery83.FieldName('MANL').AsBoolean;
        ABSTable16.FieldName('CAT').AsInteger:=
            ABSQuery83.FieldName('CAT').AsInteger;
        qtrl:=qtrl-1;
        ABSTable16.Post;
        ABSQuery83.Next;
    end;
if qtrl>0 then ShowMessage('ABSQuery83_2 errore eof');
// a questo punto la T16 contiene i dati di una disciplina

    RnkMax:=ABSTable16.RecordCount; //il valore del ranking parte dal
numero di partecipanti
    rn:=RnkMax;

    ABSTable16.First;
//ricava TMAX e il cod. disciplina dalla T4
    dp:= ABSTable16.FieldName('CODDSPL').AsInteger;
    trovato:=Form2.ABSTable4.Locate('CODDSPL',dp,[]);
    TDateTimeField(Form2.ABSTable4.FieldName('TMAX')).
        DisplayFormat:='nn:ss,zzz';

    if trovato =True then
tmx:=Form2.ABSTable4.FieldName('TMAX').AsDateTime
    else
        begin
            ShowMessage('ERRORE, Disciplina o TMAX non trovato per la
disciplina : '+inttostr(dp));
            Exit;
        end;
    while not ABSTable16.eof do //un record per volta colleziona i dati

```

di un atleta

```
begin
  ABSTable16.Edit;
  t1:=ABSTable16.FieldByName('TAME1').AsDateTime;
  t2:=ABSTable16.FieldByName('TBME2').AsDateTime;
  DecodeTime(t1,h1,m1,s1,z1);
  t1n:=h1+m1+s1+z1;
  DecodeTime(t2,h2,m2,s2,z2);
  t2n:=h2+m2+s2+z2;
  p1:=ABSTable16.FieldByName('PEN').AsDateTime;
  sql:=ABSTable16.FieldByName('DQ').AsBoolean;
  ca:= ABSTable16.FieldByName('CODATL').AsInteger;
  // controlla proprietà dei tempi
  DecodeTime(t1,h1,m1,s1,z1);
  DecodeTime(t2,h2,m2,s2,z2);
  if (h1<>0) or (h2<>0) then //se ci sono ore inserite
    begin
      ShowMessage('Impossibile classificare: presenti valori'+
        ' incompatibili per l''Atleta : '+ inttostr(ca));
      Exit;
    end;
  //tempo medio aggiungendo, se c'è la
```

penalità

```
if ((t1n=0)and(t2n<>0)) then
  begin
    ABSTable16.FieldByName('TMC').AsDateTime:= (t2+p1);
```

```
TDateTimeField(Form8.ABSTable16.FieldByName('TMC')).DisplayFormat:='nn:ss,zzz'
;
    end;
```

```
if ((t1n<>0)and(t2n=0)) then
  begin
    ABSTable16.FieldByName('TMC').AsDateTime:= (t1+p1);
```

```
TDateTimeField(Form8.ABSTable16.FieldByName('TMC')).DisplayFormat:='nn:ss,zzz'
;
    end;
```

```
    // inserisce il valore medio calcolato
  if ((t1n<>0)and(t2n<>0)) then
    begin
      tmcn:= (((t1+t2)/2)+p1);
      ABSTable16.FieldByName('TMC').AsDateTime:= tmcn;
```

```
TDateTimeField(Form8.ABSTable16.FieldByName('TMC')).DisplayFormat:='nn:ss,zzz'
;
```

```
    end;
    ABSTable16.Post;
    ABSTable16.Next; //nuovo atleta
  end;
```

```
    //A questo punto la T16 ha i dati medi per l'Assegnazione Del Ranking
    with ABSQuery85 do
  begin
    //Ordina per tempi crescenti
    Close;
    SQL.Text:='Select * from Ranking Order By "TMC"';
    ExecSQL;
```

```

Open
end;

ABSTable16.Close;
ABSTable16.EmptyTable;    //Attenzione cancella la T16
ABSTable16.Open;
ABSTable16.Edit;
ABSQuery85.Last;
ABSQuery85.First;
qtrl:=ABSQuery85.RecordCount;
while not ABSQuery85.EOF do    //AGGIORNA la T16 RANKING
begin                          //Ordinata per tempi crescenti
    ABSTable16.Insert;
    ABSTable16.FieldName('NMR').AsInteger:=
        ABSQuery85.FieldName('NMR').AsInteger;
    ABSTable16.FieldName('CODGA').AsInteger:=
        ABSQuery85.FieldName('CODGA').AsInteger;
    ABSTable16.FieldName('CODDSPL').AsInteger:=
        ABSQuery85.FieldName('CODDSPL').AsInteger;
    ABSTable16.FieldName('CODATL').AsInteger:=
        ABSQuery85.FieldName('CODATL').AsInteger;
    ABSTable16.FieldName('COGNOME').AsString:=
        ABSQuery85.FieldName('COGNOME').AsString;
    ABSTable16.FieldName('NMPET').AsInteger:=
        ABSQuery85.FieldName('NMPET').AsInteger;
    ABSTable16.FieldName('NMPOS').AsInteger:=
        ABSQuery85.FieldName('NMPOS').AsInteger;
    ABSTable16.FieldName('TAME1').AsDateTime:=
        ABSQuery85.FieldName('TAME1').AsDateTime;
    ABSTable16.FieldName('TBME2').AsDateTime:=
        ABSQuery85.FieldName('TBME2').AsDateTime;
    ABSTable16.FieldName('DQ').AsBoolean:=
        ABSQuery85.FieldName('DQ').AsBoolean;
    ABSTable16.FieldName('PEN').AsDateTime:=
        ABSQuery85.FieldName('PEN').AsDateTime;
    ABSTable16.FieldName('MANL').AsBoolean:=
        ABSQuery85.FieldName('MANL').AsBoolean;
    ABSTable16.FieldName('CAT').AsInteger:=
        ABSQuery85.FieldName('CAT').AsInteger;
    ABSTable16.FieldName('TMC').AsDateTime:=
        ABSQuery85.FieldName('TMC').AsDateTime;
    qtrl:=qtrl-1;
    ABSTable16.Post;
    ABSQuery85.Next;
end;
ABSTable16.First;    //Essendo ordinata per Tempi decrescenti
while not ABSTable16.eof do //scorre tutti i record per DQ da TMAX
begin
    ABSTable16.Edit;
    tmcn:=ABSTable16.FieldName('TMC').AsDateTime;
    ca:=ABSTable16.FieldName('CODATL').AsInteger ;
    t1:=ABSTable16.FieldName('TAME1').AsDateTime;
    t2:=ABSTable16.FieldName('TBME2').AsDateTime;
    if ABSTable16.FieldName('DQ').AsBoolean=True then
        DQSi:=True;
        DecodeTime(t1,h1,m1,s1,z1);
        tln:=h1+m1+s1+z1;
        DecodeTime(t2,h2,m2,s2,z2);
    end;
end;

```

```

t2n:=h2+m2+s2+z2;

    if (((t1n=0)and(t2n=0))or(tmcn>tmx))or (DQSi=True)) then
//Assegna direttamente DQ
    begin // e ranking alla T16
        ABSTable16.FieldName('DQ').AsBoolean:=True;
        ABSTable16.FieldName('RNK').AsInteger:=0;
        ABSTable16.FieldName('TMC').AsDateTime:=tmx;
    end else
    begin
        ABSTable16.FieldName('RNK').AsInteger :=rn;
        rn:=rn-1;
    end;
    ABSTable16.Post;
    DQSi:=False;
    ABSTable16.Next;
end;
if qtrl>0 then ShowMessage('ABSQuery85/16 errore eof');

//Ricostruisce Q85 Ordinata per Ranking decrescenti
with ABSQuery85 do
begin
    Close;
    SQL.Text:='Select * from Ranking Order By "RNK" desc';
    ExecSQL; //Ordina la Tabella 16 Ranking
    Open
end;

    ABSTable16.Close;
    ABSTable16.EmptyTable;
    ABSTable16.Open;
    ABSTable16.Edit;
    ABSQuery85.Last;
    ABSQuery85.First;
    qtrl:=ABSQuery85.RecordCount;
    while not ABSQuery85.EOF do
    begin
        ABSTable16.Insert;
        ABSTable16.FieldName('NMR').AsInteger:=
            ABSQuery85.FieldName('NMR').AsInteger;
        ABSTable16.FieldName('CODGA').AsInteger:=
            ABSQuery85.FieldName('CODGA').AsInteger;
        ABSTable16.FieldName('CODDSPL').AsInteger:=
            ABSQuery85.FieldName('CODDSPL').AsInteger;
        ABSTable16.FieldName('CODATL').AsInteger:=
            ABSQuery85.FieldName('CODATL').AsInteger;
        ABSTable16.FieldName('COGNOME').AsString:=
            ABSQuery85.FieldName('COGNOME').AsString;
        ABSTable16.FieldName('NMPET').AsInteger:=
            ABSQuery85.FieldName('NMPET').AsInteger;
        ABSTable16.FieldName('NMPOS').AsInteger:=
            ABSQuery85.FieldName('NMPOS').AsInteger;
        ABSTable16.FieldName('TAME1').AsDateTime:=
            ABSQuery85.FieldName('TAME1').AsDateTime;
        ABSTable16.FieldName('TBME2').AsDateTime:=
            ABSQuery85.FieldName('TBME2').AsDateTime;
        ABSTable16.FieldName('DQ').AsBoolean:=
            ABSQuery85.FieldName('DQ').AsBoolean;
    end;
end;

```

```

        ABSTable16.FieldName('PEN').AsDateTime:=
            ABSQuery85.FieldName('PEN').AsDateTime;
        ABSTable16.FieldName('MANL').AsBoolean:=
            ABSQuery85.FieldName('MANL').AsBoolean;
        ABSTable16.FieldName('CAT').AsInteger:=
            ABSQuery85.FieldName('CAT').AsInteger;
        ABSTable16.FieldName('TMC').AsDateTime:=
            ABSQuery85.FieldName('TMC').AsDateTime;
        ABSTable16.FieldName('RNK').AsInteger:=
            ABSQuery85.FieldName('RNK').AsInteger;
        ABSTable16.FieldName('TMC').AsDateTime:=
            ABSQuery85.FieldName('TMC').AsDateTime;

        qtrl:=qtrl-1;
        ABSTable16.Post;
        ABSQuery85.Next;
    end;

    // Aggiungo il nome e nazione
ABSTable16.First;
while not ABSTable16.EOF do
begin
    CODATL16:=ABSTable16.FieldName('CODATL').AsInteger;
    Form2.ABSTable2.Open;
    ATLsi:=Form2.ABSTable2.Locate('CODATL',CODATL16,[ ] );
    if ATLsi=False then //si blocca sul record con pari codice DSPL
        begin
            ShowMessage('Errore Atleta non trovato nell''Anagrafica');
            Exit;
        end;
    ABSTable16.Edit;

ABSTable16.FieldName('NOME').AsString:=Form2.ABSTable2.FieldName('NOME').A
sString;

ABSTable16.FieldName('NAZIONE').AsString:=Form2.ABSTable2.FieldName('NAZIO
NE').AsString;
    ABSTable16.Post;
    ABSTable16.NEXT
end;
// ci sono i DATI dei punti
    //TRASCRIZIONE su T16T STORICO

// Form2.ABSTable16T.TableName:='T16T';
Form2.ABSTable16T.Open;
    ABSTable16.First;
while not ABSTable16.EOF do
begin
    //Raccolta dati di codice
    ca:=ABSTable16.FieldName('CODATL').AsInteger;
    ga:=ABSTable16.FieldName('CODGA').AsInteger;
    dp:=ABSTable16.FieldName('CODDSPL').AsInteger;

trovato:=Form2.ABSTable16T.Locate('CODATL;CODGA;CODDSPL',VarArrayOf([intToStr(
ca),IntToStr(ga),IntToStr(dp)]),[ ] );
    if trovato=False then Form2.ABSTable16T.Insert;
    Form2.ABSTable16T.Edit;
    Form2.ABSTable16T.FieldName('CODATL').AsInteger:=ca;

```

```

Form2.ABSTable16T.FieldByName('CODGA').AsInteger:=ga;
Form2.ABSTable16T.FieldByName('CODDSPL').AsInteger:=dp;

Form2.ABSTable16T.FieldByName('COGNOME').AsString:=ABSTable16.FieldByName('COG
NOME').AsString;

Form2.ABSTable16T.FieldByName('NOME').AsString:=ABSTable16.FieldByName('NOME')
.AsString;

Form2.ABSTable16T.FieldByName('NAZIONE').AsString:=ABSTable16.FieldByName('NAZ
IONE').AsString;

Form2.ABSTable16T.FieldByName('TMC').AsDateTime:=ABSTable16.FieldByName('TMC')
.AsDateTime;

TDateTimeField(Form2.ABSTable16T.FieldByName('TMC')).DisplayFormat:='nn:ss,zzz
';

Form2.ABSTable16T.FieldByName('RNK').AsInteger:=ABSTable16.FieldByName('RNK')
.AsInteger;

Form2.ABSTable16T.FieldByName('DQ').AsBoolean:=ABSTable16.FieldByName('DQ').As
Boolean;

Form2.ABSTable16T.FieldByName('CAT').AsInteger:=ABSTable16.FieldByName('CAT')
.AsInteger;
    //Aggiunge dati mancanti
    trovato:=Form2.ABSTable4.Locate('CODDSPL',dp,[]); //Descrizione
    if trovato=True then
        begin
            Form2.ABSTable16T.FieldByName('DESCR').AsString:=
Form2.ABSTable4.FieldByName('DESCR').AsString ;
            dscrz:= Form2.ABSTable4.FieldByName('DESCR').AsString;
        end else
        begin
            ShowMessage('Errore Disciplina non trovata nell''Anagrafica
@T16T');
            Exit;
        end;
    trovato:=Form2.ABSTable5.Locate('CODGA',ga,[]); //Data Gara
    if trovato=True then
        Form2.ABSTable16T.FieldByName('DATAGA').AsDateTime:=
Form2.ABSTable5.FieldByName('DATA').AsDateTime else
        begin
            ShowMessage('Errore Gara non trovata nell''Anagrafica @T16T');
            Exit;
        end;

    Form2.ABSTable16T.Post;
    ABSTable16.Next;
end;
//determina il punteggio totale per atleta

    ABSTable16R.Close;          //Resetta la R16R prima di caricarla da
Ranking
ABSTable16R.EmptyTable;
    with ABSQueryR do
begin

```

```

        Close;
        SQL.Text:= 'INSERT INTO R16R SELECT * FROM Ranking';
        ExecSQL;
    end;

    if MessageDlg('Desideri esaminare la Classifica della Disciplina ' +
        QuotedStr(IntToStr(dp)) + #13 + 'Gara : ' + QuotedStr(IntToStr(ga)) +
        '?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
        begin
            frxReport16.Variables['desc'] := quotedstr(dscrz);
            frxReport16.Variables['max'] := quotedstr(TimeToStr(tmx));
            frxReport16.ShowReport;    //Classifiche per singole discipline
        end;    //Dati da ABSTable16 (Ranking)

        ABSQuery80.Next;    //Altra disciplina
    end;
    //quì finiscono le discipline della Gara

    Form2.ABSTable3.Next; //adesso inserisce una nuova gara
end;

if ClassDP=True then    //realizzazione report riepilogativi per classifica
per Discipline
begin
    if Application.MessageBox('Desideri esaminare la classifica di ciascuna
Disciplina'
        + #13#10 + '                                per tutte le Gare selezionate?', 'Conferma',
        MB_YESNO + MB_ICONQUESTION) = IDNO then Exit else
    begin
        ABSTable20.Open;
        Form2.ABSTable3.First;
        while not Form2.ABSTable3.eof do    //Quì inizia il giro delle gare scelte
        begin
            ga:=Form2.ABSTable3.FieldName('Codice').AsInteger;
            Form2.ABSTable3D.First;
            while not Form2.ABSTable3D.eof do    //quì inizia il giro delle
discipline scelte
            begin
                dp:=Form2.ABSTable3D.FieldName('CODDSPL').AsInteger;
                with ABSQueryR do
                begin
                    Close;
                    SQL.Text:= 'INSERT INTO repDSPL' +
                        '(SELECT * FROM T16T' +
                        ' WHERE CODGA=' + QuotedStr(IntToStr(ga)) + 'AND
CODDSPL=' + QuotedStr(IntToStr(dp)) + ' )';
                    ExecSQL;
                end;
                Form2.ABSTable3D.Next; //adesso inserisce una nuova disciplina
            end;
            Form2.ABSTable3.Next; //adesso inserisce una nuova gara
        end;
        //ho tutti i record con i TMC per le gare e discipline scelte.
    end;

    //codifica molto complessa
    with ABSQueryR do
    begin
        Close;

```



```

        SQL.Text:='Select Min(repDSPL."DATAGA") as
MinDataGa,Max(repDSPL."DATAGA") as maxDataGa from repDSPL';
        ExecSQL;
        Open;
    end;
    d1:=ABSQueryR.FieldName('MinDataGa').AsDateTime;
    d2:=ABSQueryR.FieldName('maxDataGa').AsDateTime;
    durata:=' Gare dal ' +DateToStr(d1)+' al ' +DateToStr(d2);

    with ABSQueryRep do
    begin
        Close;
        SQL.Text:='Select
CODATL,CODDSPL,DESCR,COGNOME,NOME,NAZIONE,CAT,Sum(RNK) as RNK from repDSPL'+
' GROUP BY CODATL,CODDSPL,DESCR,COGNOME,NOME,NAZIONE,CAT ORDER by
CODDSPL, RNK desc';
        ExecSQL;
        Open;
    end;                                     //report per disciplina per le gare selezionate

    frxReport8D.Variables['inifin']:=quotedstr(durata);
    frxReport8D.ShowReport;

    with ABSQueryRep do                    // punteggio complessivo
    begin
        Close;
        SQL.Text:='Select CODATL,COGNOME,NOME,NAZIONE,Sum(RNK) as RNK from
repDSPL'+
' GROUP BY CODATL,COGNOME,NOME,NAZIONE ORDER by RNK desc';
        ExecSQL;
        Open;
    end;

    frxReportCompl.ShowReport;

end;

    if (ClassDp=False) then
    begin
        Form2.ABSTable3.First;
        while not Form2.ABSTable3.eof do    //Quì inizia il giro delle gare scelte
        begin
            ga:=Form2.ABSTable3.FieldName('Codice').AsInteger;
            if MessageDlg(' Desideri esaminare la Classifica Riepilogativa '+
'della Gara : ' +QuotedStr(IntToStr(ga)) + '#13+' per tutte le discipline
praticate ?',mtconfirmation, [mbYes,mbNo], 0) = mrYes then
            begin
                with ABSQueryRep do
                begin
                    Close;
                    SQL.Text:='Select * from T16T Where CODGA = ' +
QuotedStr(IntToStr(ga)) + ' Order by CODDSPL, RNK desc';
                    ExecSQL;
                    Open;
                end;

                frxReport16T.ShowReport;    //Classifica riepilogativa gara

```

```

with ABSQuery85 do          //Dati da QueryRep
begin
    Close; //tabelle con nome numerico devono iniziare con una lettera
    SQL.Text:='Select CODATL,COGNOME,NOME,DATA GA, sum(RNK) AS TOTR NK from
T16T Where CODGA = '+'
    QuotedStr(IntToStr(ga))+' Group By CODATL,COGNOME,NOME,DATA GA Order
by TOTR NK desc';
    ExecSQL;
    Open
end;
    frxReport16T2.ShowReport; // Classifica riepilogativa punteggio
end;          //Dati da Query85 >> 16T (T16T)
    Form2.ABSTable3.Next;
end;

```

```

if ClassDp=False then
begin

    if MessageDlg(' Desideri esaminare Il Punteggio complessivo raggiunto
'+#13
        +' per le Gare selezionate ?'+#13+
        'NB. Non cambia per una sola Gara',mtconfirmation, [mbYes,mbNo], 0) =
mrYes then
    begin
        ABSTable19.Open;
        ABSTable19.Edit;
        ABSTable19.First;
        Form2.ABSTable3.First;
        while not Form2.ABSTable3.Eof do
        begin          //Colleziona i dati di tutte le gare scelte
            ga:=Form2.ABSTable3.FieldName('codice').AsInteger;
            with ABSQueryRep do
            begin
                Close;
                SQL.Text:='Select CODATL,COGNOME, NOME,NAZIONE, SUM(RNK) AS
TOTR NK from T16T Where CODGA = '
                + QuotedStr(IntToStr(ga))+'AND CAT =' +QuotedStr(IntToStr(ct))+'
Group By CODATL,COGNOME,NOME,NAZIONE ';
                ExecSQL;
                Open;
            end;
            ABSQueryRep.First;
            while not ABSQueryRep.eof do
            begin
                temp2:=ABSQueryRep.FieldName('CODATL').AsInteger;
                trovato:=ABSTable19.Locate('CODATL',temp2,[]);
                if trovato=False then
                    ABSTable19.Insert;
                    ABSTable19.Edit;

ABSTable19.FieldName('CODATL').AsInteger:=ABSQueryRep.FieldName('CODATL').
AsInteger;

ABSTable19.FieldName('COGNOME').AsString:=ABSQueryRep.FieldName('COGNOME')
.AsString;

ABSTable19.FieldName('NOME').AsString:=ABSQueryRep.FieldName('NOME').AsStr

```

```

ing;

ABSTable19.FieldName('NAZIONE').AsString:=ABSQueryRep.FieldName('NAZIONE')
.AsString;

        tt:= ABSQueryRep.FieldName('TOTRNB').AsInteger;
        tt2:=ABSTable19.FieldName('RNB').AsInteger;
        ABSTable19.FieldName('RNB').AsInteger:=tt+tt2;
        ABSTable19.Post;
        ABSQueryRep.Next;
    end;
    Form2.ABSTable3.Next;
end;
    with ABSQueryRep do
    begin
        Close;
        SQL.Text:= 'Select * from totale Order by RNB desc';
        ExecSQL;
        Open;
    end;

    frxReportTot.ShowReport;
                                //dati da QueryRep >> Totale T19
    end;
end;
end ;
    ClassDp:=False;
    RadioButton1.Checked:=False;
    RadioButton1.Color:= clRed;
    RadioButton2.Checked:=False;
    RadioButton2.Color:= clRed;
    RadioButton3.Checked:=False;
    RadioButton3.Color:= clRed;
    Shape1.Brush.Color := clRed; //Azzera e resetta la scelta Gare nell'Array
    SetLength(ArGare,0);
    SetLength(ArGare,20);
    Shape2.Brush.Color := clRed; //Azzera e resetta la scelta Discipline
nell'Array
    SetLength(ArDspl,0);
    SetLength(ArDspl,20);
    Shape3.Brush.Color:= clRed;
    ComboBox1.ItemIndex:=-1;
    ABSTable19.Close;
    ABSTable19.EmptyTable;
    ABSTable18.Close;
    ABSTable18.EmptyTable;
    Dbgrid8.DataSource.DataSet.Close;
    Dbgrid8.Refresh;
    Form8.EsciClick(self);
end;

procedure TForm8.EsciClick(Sender: TObject);    //Esci
begin
    Form2.ABSTable1.Open;
    Form2.ABSTable1.Last;
    if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
    Form2.ABSTable1.Insert;
    Form2.ABSTable1.Edit;
    Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;

```

```

    Form2.ABSTable1.Post;
    Form2.ABSTable1.Close;
    Form8.Visible:=False;
    Form8.Close;    // NB chiama il can close
    ClassFlg:=True; //per consentire di monitorizzare che Form8 è closed
    Form2.Visible:=True;
    SetLength(ArGare,0);
    SetLength(ArDspl,0);
    SetLength(ArCat,0);
    Form2.ABSTable3.Close;
    Form2.ABSTable3.EmptyTable; //resetta la tabella SelezioneGare
    Form2.ABSTable3D.Close;
    Form2.ABSTable3D.EmptyTable; //resetta la tabella SelezioneDSPL
    ABSQuery85.Close;
    ABSTable15.Close; //per consentire lo svuotamento messo su ObjInsp
Exclusive=True
    ABSTable15.EmptyTable; //resetta la tabella Classifica
    ABSTable16R.Close;
    Form2.ABSTable4.Close;
    Form2.ABSTable2.Close;
    Form2.ABSTable5.Close;
    Form2.ABSTable6.Close;
    Form2.ABSTable8.Close;
    Form2.ABSTable9.Close;
    ABSTable20.Close;
    ABSTable20.EmptyTable;
end;
end.

```

```

unit Mancanti;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type
    TForm10 = class (TForm)
        Label1: TLabel;
        Edit1: TEdit;
        Button1: TButton;
        Button2: TButton;
        Button3: TButton;
        Button4: TButton;
        Label2: TLabel;
        Button5: TButton;
        procedure Button4Click(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Button5Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form10: TForm10;
    mncti:Boolean = False;

implementation

{$R *.dfm}
uses
    Gara;
procedure TForm10.Button1Click(Sender: TObject);    //Categoria
begin
    Gara.mncat:=False;
    if Edit1.Text='' then
        begin
            Application.MessageBox('Non c'è un dato da trasferire.', 'Attenzione',
                MB_OK + MB_ICONWARNING);
            Exit;
        end else
        begin
            if MessageDlg('Il Sistema non controlla la congruità del dato. ' + #13#10 +
                ' Vuoi inserire la Categoria ?', mtConfirmation, [mbYes, mbNo],
0)=MrYes then
                begin
                    Gara.mncat:=True;
                    Gara.mancante:=StrToInt(edit1.Text);
                    ShowMessage(' Clicca sul Cognome dell''atleta interessato.')
                end else
                    Exit;

```

```
end;  
end;
```

```
procedure TForm10.Button2Click(Sender: TObject); //Pettorale
```

```
begin
```

```
  Gara.mnpet:=False;
```

```
  if Edit1.Text='' then
```

```
    begin
```

```
      Application.MessageBox('Non c'è un dato da trasferire.', 'Attenzione',  
        MB_OK + MB_ICONWARNING);
```

```
      Exit;
```

```
    end else
```

```
    begin
```

```
      if MessageDlg('Il Sistema non controlla la congruità del dato. ' + #13#10 +  
        '  Vuoi inserire il numero di Pettorale ?', mtConfirmation, [mbYes,  
mbNo], 0)=MrYes then
```

```
        begin
```

```
          Gara.mnpet:=True;
```

```
          Gara.mancante:=StrToInt(edit1.Text);
```

```
          ShowMessage(' Clicca sul Cognome dell''atleta interessato.')
```

```
        end else
```

```
          Exit;
```

```
      end;
```

```
end;
```

```
procedure TForm10.Button3Click(Sender: TObject); //Posizione
```

```
begin
```

```
  Gara.mnpos:=False;
```

```
  if Edit1.Text='' then
```

```
    begin
```

```
      Application.MessageBox('Non c'è un dato da trasferire.', 'Attenzione',  
        MB_OK + MB_ICONWARNING);
```

```
      Exit;
```

```
    end else
```

```
    begin
```

```
      if MessageDlg('Il Sistema non controlla la congruità del dato. ' + #13#10 +  
        '  Vuoi inserire il numero di Posizione ?', mtConfirmation, [mbYes,  
mbNo], 0)=MrYes then
```

```
        begin
```

```
          Gara.mnpos:=True;
```

```
          Gara.mancante:=StrToInt(edit1.Text);
```

```
          ShowMessage(' Clicca sul Cognome dell''atleta interessato.')
```

```
        end else
```

```
          Exit;
```

```
      end;
```

```
end;
```

```
procedure TForm10.Button4Click(Sender: TObject); //Esci
```

```
begin
```

```
  Form6.checkbox2.checked:=False;
```

```
  Form10.Hide;
```

```
end;
```

```
procedure TForm10.Button5Click(Sender: TObject); //MANLEVA
```

```
begin
```

```

Gara.mnmanl:=False;
if Edit1.Text='' then
  begin
    Application.MessageBox('Non c'è un dato da trasferire.', 'Attenzione',
      MB_OK + MB_ICONWARNING);
    Exit;
  end else
  begin
    if MessageDlg('Il Sistema non controlla la congruità del dato. ' + #13#10 +
      '  Vuoi modificare la Manleva ?', mtConfirmation, [mbYes, mbNo],
0)=MrYes then
      begin
        if Edit1.Text='Si' then
          begin
            Gara.manleva:=True ;
            Gara.mnmanl:=True;
          end else
            if Edit1.Text='No' then
              begin
                Gara.manleva:=False;
                Gara.mnmanl:=True;
              end else
                begin
                  Gara.manleva:=False;
                  Gara.mnmanl:=False;
                  ShowMessage(' Errore di inserimento (Si/No)') ;
                  Exit;
                end;
              ShowMessage(' Clicca sul Cognome dell''atleta interessato.')
            end else
              Exit;
          end;
        end;
      end;
    end.

```

```

unit Tempi;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, System.UITypes;

type
    TForm9 = class (TForm)
        Edit1: TEdit;
        Button1: TButton;
        Label1: TLabel;
        Label2: TLabel;
        Edit2: TEdit;
        Edit3: TEdit;
        Button3: TButton;
        RadioButton1: TRadioButton;
        RadioButton2: TRadioButton;
        RadioButton3: TRadioButton;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Label6: TLabel;
        procedure Button1Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form9: TForm9;

implementation

{$R *.dfm}
{$D+}
uses
    Gara;
procedure TForm9.Button1Click(Sender: TObject); //ACCETTA
begin
    inserisco7:=False; //altrimenti se proviene da un precedente inserimento
    non consente
    if (Edit1.Text='') and (Edit2.Text='') then // 0 min 0 sec 0 ms
        begin
            ShowMessage('Nulla da accettare');
            if MessageDlg('Vuoi Uscire ?',
                mtconfirmation, [mbYes,mbNo], 0) = mrYes then
                begin
                    hide;
                    Exit;
                end
            end;
    if (Edit2.Text<>'') and (Length(Edit2.Text)<6) then

```



```

begin
    ShowMessage('I secondi devono avere il formato YY,ZZZ');
    Exit;
end;

Gara.rab1:=False; //i rab 1 o 2 sono flag se TempoA o B
Gara.rab2:=False; // rab3 = penalità
Gara.rab3:=False;
if (Edit1.Text<>'') and (Edit2.Text='') then // 1 min 0 sec
begin
    Edit3.Text:='';
    Edit3.Text:='00:'+Edit1.Text+':00,000';
    // Edit1.Text:='';
    // Edit2.Text:='';

end;
if (Edit1.Text='') and (Edit2.Text<>'') then // 0 min 1 sec
begin
    Edit3.Text:='';
    Edit3.Text:= '00:00:'+Edit2.Text;
    //Edit1.Text:='';
    //Edit2.Text:='';
end;
if (Edit1.Text<>'') and (Edit2.Text<>'') then // 1 min 1 sec
begin
    Edit3.Text:='';
    Edit3.Text:='00:'+Edit1.Text+':'+Edit2.Text;
    //Edit1.Text:='';
    //Edit2.Text:='';
end;

Gara.tempo:=Edit3.Text;
if RadioButton1.Checked=True then
    Gara.rab1:=True;
if RadioButton2.Checked=True then
    Gara.rab2:=True;
if RadioButton3.Checked=True then
    Gara.rab3:=True;
// RadioButton1.Checked:=True;
if Edit3.Text<>' ' then //se c'è un tempo da caricare
ShowMessage('Clicca sul Cognome dell''atleta interessato.') else
begin
    Application.MessageBox('Errore non previsto', 'Attenzione',
        MB_OK + MB_ICONWARNING);
    exit;
end;

end;

procedure TForm9.Button3Click(Sender: TObject); // Esci
begin
    Form9.hide;
end;

procedure TForm9.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if (Key=#13) AND (length(edit2.text)<6) then

```

```
    showmessage('Il formato deve essere YY,ZZZ ');  
end;  
end.
```

```

unit aiuto;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type
    TForm7 = class(TForm)
        mmol: TMemo;
        Button1: TButton;
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form7: TForm7;

implementation
    uses
        Gara;
    {$R *.dfm}
    {$D+}
    procedure TForm7.Button1Click(Sender: TObject);
    begin
        Form7.visible:=False;
    end;

end.

```