

```
1: unit Gara;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Mask, Vcl.DBCtrls,
8:   Vcl.ExtCtrls, Vcl.Grids, Vcl.DBGrids, Data.DB, ABSMain, Vcl.ComCtrls,
9:   Vcl.Imaging.jpeg, Vcl.Buttons, frxClass, frxDBSet, frxExportPDF,
10:  Datasnap.Provider, Datasnap.DBClient;
11:
12: type
13:   TForm6 = class(TForm)
14:     Button1: TButton;
15:     Button2: TButton;
16:     Button3: TButton;
17:     Button4: TButton;
18:     dbgrd72: TDBGrid;
19:     dbnvgr7: TDBNavigator;
20:     ds7: TDataSource;
21:     ds72: TDataSource;
22:     AbsQuery72: TABSQuery;
23:     cbb72: TComboBox;
24:     Label7: TLabel;
25:     Timer1: TTimer;
26:     Label8: TLabel;
27:     shp1: TShape;
28:     Label9: TLabel;
29:     dbgrd7: TDBGrid;
30:     dbgrd74: TDBGrid;
31:     ds74: TDataSource;
32:     Label1: TLabel;
33:     Edit1: TEdit;
34:     Edit2: TEdit;
35:     ABSQuery7: TABSQuery;
36:     Button5: TButton;
37:     ABSQuery74: TABSQuery;
38:     img2: TImage;
39:     cbb74: TComboBox;
40:     Label2: TLabel;
41:     rb1: TRadioButton;
42:     rb2: TRadioButton;
43:     btn1: TBitBtn;
44:     img3: TImage;
45:     Label3: TLabel;
46:     Label4: TLabel;
47:     frxdbdtst13: TfrxDBDataset;
48:     ABSQuery73: TABSQuery;
49:     ABSTable13: TABSTable;
50:     ds13: TDataSource;
51:     frxrprt7: TfrxReport;
52:     frxPDFExport7: TfrxPDFExport;
53:     Button6: TButton;
54:     Edit3: TEdit;
55:     Button7: TButton;
56:     ABSTable14: TABSTable;
57:     Button8: TButton;
58:     Bevel1: TBevel;
59:     Label5: TLabel;
60:     Label6: TLabel;
61:     Button9: TButton;
62:     img1: TImage;
63:     Edit4: TEdit;
64:     Label10: TLabel;
65:     Button10: TButton;
66:     frxReport14: TfrxReport;
```

```

68:     ABSQueryDUP: TABSQuery;
69:     ABSQueryDUP2: TABSQuery;
70:     procedure Button1Click(Sender: TObject);
71:     procedure FormCreate(Sender: TObject);
72:     procedure dbgrd72CellClick(Column: TColumn);
73:     procedure Button2Click(Sender: TObject);
74:     procedure dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);
75:     procedure cbb72Select(Sender: TObject);
76:     procedure Timer1Timer(Sender: TObject);
77:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
78:     procedure Button5Click(Sender: TObject);
79:     procedure cbb74Change(Sender: TObject);
80:     procedure rb2Click(Sender: TObject);
81:     procedure rb1Click(Sender: TObject);
82:     procedure Button6Click(Sender: TObject);
83:     procedure btn1Click(Sender: TObject);
84:     procedure Edit1Change(Sender: TObject);
85:     procedure Button3Click(Sender: TObject);
86:     procedure Button4Click(Sender: TObject);
87:     procedure Button7Click(Sender: TObject);
88:     procedure Button8Click(Sender: TObject);
89:     procedure ABSTable14BeforeInsert(DataSet: TDataSet);
90:     procedure ABSTable14BeforeDelete(DataSet: TDataSet);
91:     procedure Button9Click(Sender: TObject);
92:     procedure dbgrd7CellClick(Column: TColumn);
93:     procedure GetText(Sender: TField; var Text: string; DisplayText: Boolean);
94:     procedure Button10Click(Sender: TObject);
95:     private
96:     { Private declarations }
97:     public
98:     { Public declarations }
99:     end;
100:         //NOTA BENE l'entry del codice gara non triggera alcun evento
101:     var
102:         Form6: TForm6;
103:         GaFlg:Boolean =True;
104:         inserisco7:Boolean=False;
105:         ATL:Integer;
106:         scelta:Boolean=False; // se si sceglie con la combobox > True
107:         filtGa:Boolean=False; // idem se filtra Gara
108:         CanClose:Boolean=True;
109:         LastGa:SmallInt;
110:         numgara:Integer;
111:         tempo:string;
112:         rab1,rab2:Boolean;
113:     implementation
114:         uses
115:             Console1,aiuto,Tempi ;
116:         {$R *.dfm}
117:         {$D+}
118:     procedure TForm6.FormCreate(Sender: TObject); //Crea il form
119:     begin
120:         Form2.ABSTable7.TableName:='Gara';
121:         Form2.ABSTable7.Open; //FORMATTARE CAMPI DBGRID
122:         TDateTimeField(Form2.ABSTable7.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
123:         TDateTimeField(Form2.ABSTable7.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
124:         TDateTimeField(Form2.ABSTable7.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
125:         TBooleanField(Form2.ABSTable7.FieldByName('DQ')).DisplayValues:='Yes;No';
126:         ABSTable14.TableName:='Corrente';
127:         ABSTable14.Open;
128:         TDateTimeField(ABSTable14.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
129:         TDateTimeField(ABSTable14.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
130:         TDateTimeField(ABSTable14.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
131:         TBooleanField(ABSTable14.FieldByName('DQ')).DisplayValues:='Yes;No';
132:         GaFlg:=False;
133:         ds7.DataSet:=Form2.ABSTable7;//aggiunto per evitare circular DataLink error
134:         Form2.ABSTable7.Edit;

```

```

136:     Form2.ABSTable1.Open;
137:     Form2.ABSTable1.INSERT;
138:     Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
139:     Form2.ABSTable1.FieldName('what').AsString:='Accesso alle Iscrizioni';
140:     Form2.ABSTable1.FieldName('OUT').AsString:='';
141:     Form2.ABSTable1.Post;
142:     Form2.ABSTable1.Close;
143:     cbb74.Text:='Scegli';
144:     if rb1.Checked=True then
145:     begin
146:         Edit2.Visible:=False;
147:         Label1.Visible:=False;
148:     end;
149:     Form2.ABSTable5.TableName:='AnaGa';
150:     Form2.ABSTable5.Open;
151:     Form2.ABSTable5.Last;
152:     LastGa:=Form2.ABSTable5.FieldName('CODGA').AsInteger ;
153:     Label4.Caption:=IntToStr(LastGa);
154:     Button2.Click;    //Refresh
155:     Button5.Click;
156:     ATL:=0;
157:
158: end;
159:
160: procedure TForm6.GetText(Sender: TField; var Text: string; DisplayText: Boolean);
161: begin
162:     if ABSTable14.FieldName('DQ').AsBoolean then
163:         Text := 'Yes'
164:     else
165:         Text := 'No';
166:
167: end;
168:
169:
170:
171: procedure TForm6.rb1Click(Sender: TObject); //AUTOMATICA
172: begin
173:     Edit2.Visible:=False;    //toglie edit2
174:     Label1.Visible:=False;
175:     cbb74.Visible:=True;
176:     Label2.Visible:=True;
177: end;
178:
179: procedure TForm6.rb2Click(Sender: TObject); // MANUALE
180: begin
181:     if Edit1.Text='' then
182:     begin
183:         ShowMessage('Inserisci il Codice della Gara ');
184:         rb1.Checked:=True;
185:         Exit;
186:     end;
187:     cbb74.Visible:=False;
188:     Label2.Visible:=False;
189:     Edit2.Visible:=True;
190:     Label1.Visible:=True;
191:     ds74.DataSet:=Form2.ABSTable4;
192: end;
193:
194: procedure TForm6.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
195: begin
196:     if inserisco7=True then
197:     if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) = mrYes
198:     then
199:     begin
200:         CanClose:=False;
201:         Exit;
202:     end

```

```

204:      begin
205:          CanClose:=True;
206:          if inserisco7=True then //conferma il Falso di inserisco5
207:              inserisco7:=False;
208:              Button1Click(Self);
209:      end;
210: end;
211:
212:
213: procedure TForm6.Timer1Timer(Sender: TObject);    //Flashing
214: begin
215:     if Edit1.Text='' then
216:         begin
217:             if label8.Font.Color = clRed then
218:                 label8.Font.Color :=  clNavy
219:             else
220:                 label8.Font.color := clRed;
221:             end else Label8.Visible:=False;
222:     if Edit2.Text='' then
223:         begin
224:             if label11.Font.Color = clRed then
225:                 label11.Font.Color :=  clNavy
226:             else
227:                 label11.Font.color := clRed;
228:             end else Label11.Visible:=False;
229:     //condizioni per garantire che label2 sia invisibile, altrimenti il colore vince
230:     if ((cbb74.ItemIndex=0) or (cbb74.Text='Scegli'))and (cbb74.Visible=True) then
231:         begin
232:             if label2.Font.Color = clRed then
233:                 label2.Font.Color :=  clNavy
234:             else
235:                 label2.Font.color := clRed;
236:                 Label2.Visible:=True;
237:             end else Label2.Visible:=False;
238:
239:     end;
240:
241: procedure TForm6.ABSTable14BeforeDelete(DataSet: TDataSet);
242: begin
243:     if ds7.DataSet=ABSTable14 then
244:         begin
245:             ShowMessage('Prima di rimuovere un record premi "Refresh" '+#13+
246:                 'ed identifica il record con il suo codice');
247:             Abort;
248:             Exit;
249:         end;
250:     end;
251:
252: procedure TForm6.ABSTable14BeforeInsert(DataSet: TDataSet);
253: begin
254:     if ds7.DataSet=ABSTable14 then
255:         begin
256:             ShowMessage('Per inserire nuovi record premi "Refresh" '+#13+
257:                 'o esegui l''inserimento automatico');
258:             Abort;
259:             Exit;
260:         end;
261:     end;
262:
263: procedure TForm6.btn1Click(Sender: TObject);
264: begin
265:     Form7.Show;
266: end;
267:
268: procedure TForm6.Button1Click(Sender: TObject);    //Esci
269:
270: begin
271: You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

272:     if filtGa =True then
273:     if MessageDlg('Devi sincronizzare la tabella delle iscrizioni ?'
274: , mtconfirmation, [mbYes,mbNo], 0) = mrYes then Exit;
275:     filtGa:=False;
276:     if inserisco7=True then //se ha inserito un record
277:     begin
278:         FormCloseQuery(Self,canclose);
279:     end
280:     else
281:     begin
282:         Form2.ABSTable7.Close;
283:         Form6.Close; // NB chiama il can close
284:         GaFlg:=True; //per consentire di monitorizzare che Form6 è closed
285:         Form2.Visible:=True;
286:         Edit1.Text:=''; //rimuove il codice della gara
287:         Edit2.Text:=''; // '' della DSPL
288:         Form2.ABSTable10.Close; //Chiude non cancella
289:         Form2.ABSTable11.Close;
290:         Form2.ABSTable12.Close;
291:         Form6.ABSTable13.Close;
292:         Form6.ABSTable13.EmptyTable;
293:         Form6.ABSTable14.Close;
294:         Form6.ABSTable14.EmptyTable;
295:         Form2.ABSTable2.Close;
296:         Form2.ABSTable4.Close;
297:         Form2.ABSTable5.Close;
298:         Form2.ABSTable6.Close;
299:         Form2.ABSTable8.Close;
300:         Form2.ABSTable9.Close;
301:     end;
302:
303: end;
304:
305: procedure TForm6.Button2Click(Sender: TObject); //Refresh dbgrid Atleti
306: begin
307:     ds72.DataSet:=AbsQuery72;
308:     with AbsQuery72 do
309:     begin
310:         Close; //inserisce una variabile nel query
311:         SQL.Text := 'Select * from AnaAtl';
312:         ExecSql;
313:         Open;
314:     end;
315: end;
316:
317: procedure TForm6.Button3Click(Sender: TObject); //Rapporto
318: var
319: CODGAn:Integer;
320: DSPLsi,ATLsi:Boolean;
321: CODDSPL13,CODATL13:Integer;
322: begin
323:     Form2.ABSTable4.Open;
324:     //Form2.ABSTable4.Edit;
325:     Form2.ABSTable2.Open;
326:     //Form2.ABSTable2.Edit;
327:     CODGAn:=StrToInt(inputBox('Rapporto Iscrizioni','Inserire il codice della Gara',Edit1.Text));
328:     ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
329:     with ABSQuery7 do
330:     begin
331:         Close; //porto su DBGRID solo la gara scelta con tutti gli iscritti
332:         SQL.Text := 'Select * from Gara where CODGA = '+QuotedStr(IntToStr(CODGAn))+ '
ORDER BY CODATL Asc, CODDSPL Asc';
333:         ExecSql;
334:         Open; //Estrae i dati di iscrizione solo per una gara
335:         First;
336:         Form6.ABSTable13.TableName:='Maestra'; // Lo ricostruisce da cleaned
337: You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

338:         Form6.ABSTable13.EmptyTable;
339:         Form6.ABSTable13.Open;
340:         TBooleanField(ABSTable13.FieldByName('DQ')).DisplayValues:='Yes;No';
341:         TDateTimeField(ABSTable13.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
342:         TDateTimeField(ABSTable13.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
343:         TDateTimeField(ABSTable13.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
344:         Form6.ABSTable13.Edit;           // Crea Tabella Master per il report = Ma
estra
345:         while (not Eof) do //sempre sotto ABSQuery7
346:         begin                // copia il Query in una Table
347:             Form6.ABSTable13.Insert;
348:             Form6.ABSTable13.FieldByName('CODGA').AsInteger:=ABSQuery7.FieldByName('CODG
A').AsInteger;
349:             Form6.ABSTable13.FieldByName('CODDSPL').AsInteger:=ABSQuery7.FieldByName('CO
DDSPL').AsInteger;
350:             Form6.ABSTable13.FieldByName('CODATL').AsInteger:=ABSQuery7.FieldByName('COD
ATL').AsInteger;
351:             Form6.ABSTable13.FieldByName('COGNOME').AsString:=ABSQuery7.FieldByName('COG
NOME').AsString;
352:             Form6.ABSTable13.FieldByName('NMPET').AsInteger:=ABSQuery7.FieldByName('NMPE
T').AsInteger;
353:             Form6.ABSTable13.FieldByName('NMPOS').AsInteger:=ABSQuery7.FieldByName('NMPO
S').AsInteger;
354:             Form6.ABSTable13.FieldByName('TAME1').AsDateTime:=ABSQuery7.FieldByName('TAM
E1').AsDateTime;
355:             Form6.ABSTable13.FieldByName('TBME2').AsDateTime:=ABSQuery7.FieldByName('TBM
E2').AsDateTime;
356:             Form6.ABSTable13.FieldByName('DQ').AsBoolean:=ABSQuery7.FieldByName('DQ').As
Boolean;
357:             Form6.ABSTable13.FieldByName('PEN').AsDateTime:=ABSQuery7.FieldByName('PEN')
.AsDateTime;
358:             Form6.ABSTable13.FieldByName('MANL').AsBoolean:=ABSQuery7.FieldByName('MANL'
).AsBoolean;
359:             Form6.ABSTable13.FieldByName('CAT').AsInteger:=ABSQuery7.FieldByName('CAT').
AsInteger;
360:             Form6.ABSTable13.Post;
361:             ABSQuery7.Next;
362:         end;
363:
364:         end;           //aggiungo i campi DESCR e TMAX
365:         ABSTable13.First;
366:         while not ABSTable13.EOF do
367:         begin
368:             CODDSPL13:=ABSTable13.FieldByName('CODDSPL').AsInteger;
369:             DSPLsi:=Form2.ABSTable4.Locate('CODDSPL',CODDSPL13,[ ]);//si blocca sul record con p
ari codice DSPL
370:             if DSPLsi=False then ShowMessage('Valore non trovato'); //NB nel locate il valore d
a ricercare è variant (non stringa)
371:             ABSTable13.Edit;
372:             ABSTable13.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName('DESCR').AsStr
ing;
373:             ABSTable13.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('TMAX').AsDat
eTime;
374:             ABSTable13.Post;
375:             ABSTable13.NEXT
376:         end;
377:         // Aggiungo il nome e nazione e il telefono
378:         ABSTable13.First;
379:         while not ABSTable13.EOF do
380:         begin
381:             CODATL13:=ABSTable13.FieldByName('CODATL').AsInteger;
382:             ATLsi:=Form2.ABSTable2.Locate('CODATL',CODATL13,[ ]);//si blocca sul record con pari
codice DSPL
383:             if ATLsi=False then ShowMessage('Valore non trovato');
// NB nel locate il valore da ricercare è variant (non stringa)
384:             ABSTable13.Edit;
385:             ABSTable13.FieldByName('NOME').AsString:=Form2.ABSTable2.FieldByName('NOME').AsStrin

```

```

386:         ABSTable13.FieldName('NAZIONE').AsString:=Form2.ABSTable2.FieldName('NAZIONE').A
sString;
387:         ABSTable13.FieldName('TLF').AsString:=Form2.ABSTable2.FieldName('TLF').AsString;
388:         ABSTable13.Post;
389:         ABSTable13.NEXT
390:     end;
391:     Form6.frxrprt7.ShowReport;
392: end;
393:
394: procedure TForm6.Button4Click(Sender: TObject); //Stampa Report
395: begin
396:     Form6.frxrprt7.Print;
397: end;
398:
399: procedure TForm6.Button5Click(Sender: TObject); //Refresh tabella iscrizioni
400: begin
401:     dbgrd7.Columns[0].FieldName:='NMREC'; //applicare il campo relativo al cambio di Dataset
402:     Label6.Caption:='Dati relativi a tutte le Gare';
403:     ds7.DataSet:=Form2.ABSTable7;
404: end;
405:
406: procedure TForm6.Button6Click(Sender: TObject); // Filtro una gara          LAVORI IN CORSO
CODICE DA RIVEDERE
407: var
408:     quanti,btscelta :Integer;
409:     cosa:TStringList;
410:     ca,cd,ga:Integer;
411: begin
412:     if Edit3.Text='' then
413:     begin
414:         ShowMessage('Inserisci il codice della Gara nella casella a destra !');
415:         Exit;
416:     end;
417:     Form6.ABSTable14.TableName:=('Corrente');
418:     Form6.ABSTable14.Open;
419:     Form6.ABSTable14.Edit;
420:     quanti:=ABSTable14.RecordCount;
421:     filtGa:=True;
422:     if quanti>0 then //Continua solo se non esiste già il buffer
423:     begin
424:         numgara:=ABSTable14.FieldName('CODGA').AsInteger;
425:         if numgara<> StrToInt(Edit3.Text) then //numero gara residente DIVERSO dal richiest
o
426:         begin
427:             if MessageDlg('La tabella contiene dati per una Gara diversa '+#13#10+
428:                 'e con codice : '+ IntToStr(numgara)+' Desideri cancellarli ?', mtconfirmatio
n,
429:                 [mbYes,mbNo], 0) = mrYes then
430:                 begin
431:                     Button7Click(self);
432:                     Edit3.Text:='';
433:                     Exit;
434:                 end else
435:                 begin //Gara diversa da quella richiesta non cancellata
436:                     ShowMessage('Puoi esaminare una gara per volta, ripeti la scelta !');
437:                     Edit3.Text:='';
438:                     Exit;
439:                 end;
440:             end else
441:             begin
442:                 btscelta := MessageDlg('La Tabella per la Gara n. '+IntToStr(numgara)+' ha '+Int
ToStr(quanti)+
443:                 ' dati.'+#13+' Vuoi usarla(Yes) o cancellarla(No) o rinunciare(Cancel) ? ',mtCu
stom,
444:                 [mbYes,mbNo,mbCancel], 0);
445:                 if btscelta = mrYes then //usa il precedente buffer e continua
446:                 begin

```



## Dataset

```

448:         ds7.DataSet:=ABSTable14;
449:         Label6.Caption:='Dati relativi ad una Gara';
450:         ShowMessage('E'' possibile effettuare modifiche');
451:         Edit3.Text:=''; //Punto in cui si è scelta la GARA da Aggiornare
452:         Exit;
453:     end;
454:     if btscelta = mrNo then // cancella il buffer per rifarlo
455:     begin
456:         Button7Click(Self);
457:         Edit3.Text:='';
458:         exit;
459:     end;
460:     if btscelta = mrCancel then //esce dalla procedura
461:     begin
462:         Edit3.Text:='';
463:         Exit;
464:     end;
465: end;
466: end;
467:
468: dbgrd7.Columns[0].FieldName:='NM'; //applicare il campo relativo al cambio di Dataset
469: ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
470: with ABSQuery7 do
471: begin
472:     Close; //porto su DBGRID solo la gara scelta con tutti gli iscritti
473:     SQL.Text := 'Select * from Gara where CODGA = '+QuotedStr(Edit3.Text)+ ' ORDER
BY CODATL Asc, CODDSPL Asc';
474:     ExecSql;
475:     Open; //Estrae i dati di iscrizione solo per una gara
476:     First;
477:     while (not Eof) do //sempre sotto ABSQuery7
478:     begin // copia il Query in una Table 14 Buff
er gara corrente
479:         Form6.ABSTable14.Insert;
480:         Form6.ABSTable14.FieldName('NM').AsInteger:=ABSQuery7.FieldName('NMREC')
.AsInteger;
481:         Form6.ABSTable14.FieldName('CODGA').AsInteger:=ABSQuery7.FieldName('CODG
A').AsInteger;
482:         Form6.ABSTable14.FieldName('CODDSPL').AsInteger:=ABSQuery7.FieldName('CO
DDSPL').AsInteger;
483:         Form6.ABSTable14.FieldName('CODATL').AsInteger:=ABSQuery7.FieldName('COD
ATL').AsInteger;
484:         Form6.ABSTable14.FieldName('COGNOME').AsString:=ABSQuery7.FieldName('COG
NOME').AsString;
485:         Form6.ABSTable14.FieldName('NMPET').AsInteger:=ABSQuery7.FieldName('NMPE
T').AsInteger;
486:         Form6.ABSTable14.FieldName('NMPOS').AsInteger:=ABSQuery7.FieldName('NMPO
S').AsInteger;
487:         Form6.ABSTable14.FieldName('TAME1').AsDateTime:=ABSQuery7.FieldName('TAM
E1').AsDateTime;
488:         Form6.ABSTable14.FieldName('TBME2').AsDateTime:=ABSQuery7.FieldName('TBM
E2').AsDateTime;
489:         Form6.ABSTable14.FieldName('DQ').AsBoolean:=ABSQuery7.FieldName('DQ').As
Boolean;
490:         Form6.ABSTable14.FieldName('PEN').AsDateTime:=ABSQuery7.FieldName('PEN')
.AsDateTime;
491:         Form6.ABSTable14.FieldName('MANL').AsBoolean:=ABSQuery7.FieldName('MANL')
.AsBoolean;
492:         Form6.ABSTable14.FieldName('CAT').AsInteger:=ABSQuery7.FieldName('CAT').
AsInteger;
493:         Form6.ABSTable14.Post;
494:         ABSQuery7.Next;
495:     end;
496: end;
497: Label6.Caption:='Dati relativi ad una Gara';
498: ds7.DataSet:=ABSTable14; //MOSTRA LA TABELLA GARA CORRENTE

```



```

500:      cosa:=TStringList.Create;
501:      cosa.Clear;
502:      ABSTable14.First;
503:      while not ABSTable14.Eof do
504:      begin
505:          if ABSTable14.FieldName('CAT').AsInteger=0 then
506:          begin
507:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
508:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
509:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
510:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+inttostr(ca)+'- Disciplina:
'+inttostr(cd)+'> Categoria');
511:          end;
512:          if ABSTable14.FieldName('NMPET').AsInteger=0 then
513:          begin
514:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
515:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
516:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
517:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+inttostr(ca)+'- Disciplina: '
+inttostr(cd)+'> Pettorale');
518:          end;
519:          if ABSTable14.FieldName('NMPOS').AsInteger=0 then
520:          begin
521:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
522:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
523:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
524:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+inttostr(ca)+'- Disciplina: '
+inttostr(cd)+'> Posizione');
525:          end;
526:          ABSTable14.Next;
527:      end;
528:      if cosa.Count>0 then
529:      begin
530:          cosa.SaveToFile('.\mancanti.txt');
531:          if (cosa.Count<50) then
532:              ShowMessage('Codici Atleti con dati mancanti-Annotarli ' + #13+#13+ cosa.Text+#13+
533:              ' Dati salvati in "mancanti.txt") else
534:              ShowMessage('La lista degli Atleti con dati mancanti è salvata in "Mancanti.txt".
535:      ');
536:
537:      //Correzione DATI Mancanti
538:      if MessageDlg('Desideri fornire ora i dati mancanti essenziali ?'+#13+#13
539:      + 'Ricorda di caricare il dato e sincronizzare',mtconfirmation, [mbYes,mbNo], 0)
540:      = mrYes then
541:      begin
542:          ds7.DataSet:=ABSQuery7;
543:          with ABSQuery7 do
544:          begin
545:              Close;
546:              SQL.Text := 'Select * from Corrente where CAT = '+QuotedStr(IntToStr(0)) ;
547:              ExecSql;
548:              Open;
549:          end;
550:      end;
551:      ShowMessage('E'' possibile effettuare modifiche');
552:      Edit3.Text:='';
553:      cosa.Clear;
554:  end;
555:
556:
557: procedure TForm6.Button7Click(Sender: TObject); // Resetta la tabella della Gara corrente
558: begin
559:     if MessageDlg('Stai per azzerare i dati memorizzati nel Buffer. Vuoi continuare ?',
560:     mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
561:     begin
562:

```

```

563:         ABSTable14.EmptyTable;
564:         ShowMessage('Tutti i dati di iscrizione contenuti nel buffer sono stati cancellat
i, '
565:             + #13+ ' i dati sincronizzati sono presenti nel Database');
566:         end;
567:         ds7.DataSet:=Form2.ABSTable7;
568:     end;
569:
570: procedure TForm6.Button8Click(Sender: TObject);    //Sincronizza buffer con Gara
571: var
572:     doppio:Boolean;
573:     cerco:integer;
574: begin
575:     if filtGa=False then
576:     begin
577:         ShowMessage('Non c''è nulla da sincronizzare. ');
578:         Exit;
579:     end;
580:     if MessageDlg('Confermi di voler aggiornare i Dati Base ?', mtconfirmation,
581:         [mbYes,mbNo], 0) = mrNo then Exit;
582:
583:     Form2.ABSTable7.Open;
584:     ABSTable14.Open;
585:     ABSTable14.First;
586:     ABSTable14.Edit;
587:     while not ABSTable14.eof do
588:     begin
589:         cerco:=ABSTable14.FieldByName('NM').AsInteger;
590:         doppio:=Form2.ABSTable7.Locate('NMREC',cerco,[]);
591:         if doppio=True then
592:         begin //record preesistente da aggiornare
593:             Form2.ABSTable7.Edit;
594:             Form2.ABSTable7.FieldByName('NMPET').AsInteger:=ABSTable14.FieldByName('NMPET').As
Integer;
595:             Form2.ABSTable7.FieldByName('NMPOS').AsInteger:=ABSTable14.FieldByName('NMPOS').As
Integer;
596:             Form2.ABSTable7.FieldByName('TAME1').AsDateTime:=ABSTable14.FieldByName('TAME1').A
sDateTime;
597:             Form2.ABSTable7.FieldByName('TBME2').AsDateTime:=ABSTable14.FieldByName('TBME2').A
sDateTime;
598:             Form2.ABSTable7.FieldByName('DQ').AsBoolean:=ABSTable14.FieldByName('DQ').AsBoolea
n;
599:             Form2.ABSTable7.FieldByName('PEN').AsDateTime:=ABSTable14.FieldByName('PEN').AsDat
eTime;
600:             Form2.ABSTable7.FieldByName('MANL').AsBoolean:=ABSTable14.FieldByName('MANL').AsBo
olean;
601:             Form2.ABSTable7.FieldByName('CAT').AsInteger:=ABSTable14.FieldByName('CAT').AsInte
ger;
602:             Form2.ABSTable7.Post;
603:         end else
604:         begin
605:             ShowMessage('ERRORE ! Il record con il codice '+IntToStr(cerco)+
606:                 ' non esiste nella Tabella Gare Generale. ');
607:         end;
608:         ABSTable14.Next;
609:     end;
610:     ShowMessage('La tabella Gare Generale è stata aggiornata.'+#13+
611:         'Puoi cliccare "Azzerà" per cancellare il buffer. ');
612: end;
613:
614: procedure TForm6.Button9Click(Sender: TObject);
615: begin
616:     if filtGa=False then
617:     begin
618:         ShowMessage('Non c''è alcuna Gara selezionata');
619:         Exit;
620:     end;

```

```

622: end;
623:
624:
625: procedure TForm6.Button10Click(Sender: TObject); //Rapporto iscritti
626: begin
627:     if Edit3.Text='' then
628:     begin
629:         ShowMessage('Inserisci il codice della Gara nella casella a destra !');
630:         Exit;
631:     end;
632:
633:     frxReport14.ShowReport;
634:
635: end;
636:
637: //Inserimento atleti nella gara cliccando su un Atleta
638: procedure TForm6.dbgrd72CellClick(Column: TColumn);
639: var
640:     Nom,Cogn:string;
641:     volte:smallInt;
642:     Sorge:TABSTable;
643:     Rect2:Boolean;
644:     trovato:Boolean;
645:     n:Integer;
646: begin
647:     if Edit1.Text<>' ' then
648:     begin
649:         // o usa la Tab2=ATL o Query2 = ATL da cognome
650:         Form2.ABSTable2.Open;
651:         if scelta=False then
652:         begin
653:             ATL:= dbgrd72.Fields[0].AsInteger;//CODATL dal campo 0 della tabella di cui ho cliccat
o una colonna
654:         end else ATL:=AbsQuery72.FieldByName('CODATL').AsInteger; //prende ATL dalla scelta
655:
656:         if Form2.ABSTable2.RecordCount >0 then
657:             Rect2:=Form2.ABSTable2.Locate('CODATL',ATL,[]) else
658:             begin
659:                 ShowMessage('La tabella degli Atleti è vuota. ');
660:                 Exit;
661:             end;
662:         if Rect2=False then
663:         begin
664:             ShowMessage('Errore Imprevisto: l''Atleta non esiste nel DataBase - Controlla');
665:             Exit;
666:         end;
667:         Cogn:= Form2.ABSTable2.FieldByName('COGNOME').AsString;
668:         Nom:= Form2.ABSTable2.FieldByName('NOME').AsString;
669:
670:         if rb1.Checked=True then //automatica
671:         begin
672:             if MessageDlg('Confermi l''iscrizione automatica ?',
673:                 mtconfirmation, [mbYes,mbNo], 0) = mrYes then
674:             begin
675:                 if cbb74.ItemIndex=-1 then //Att. cbb.Text='xx' funge dopo una scelta
676:                 begin
677:                     ShowMessage('Scegli il Gruppo di iscrizione');
678:                     Exit;
679:                 end;
680:                 end;
681:                 //SCELTA AUTOMATICA dopo la scelta della Gara e del Gruppo
682:                 if MessageDlg('Confermi l''iscrizione AUTOMATICA di '+Nom+ ' ' + Cogn+
683:                     ' alle discipline del gruppo '+cbb74.Text,
684:                     mtconfirmation, [mbYes,mbNo], 0) = mrNo then
685:                 begin
686:                     Button2.Click;
687:                     Exit;
688:                 end;

```

```
689:      begin
690:      if cbb74.Text='Base' then
691:      begin
692:      Sorge:=Form2.ABSTable10;
693:      Sorge.Open;
694:      end;
695:
696:      if cbb74.Text='GruppoA' then
697:      begin
698:      Sorge:=Form2.ABSTable11;
699:      Sorge.Open;
700:      end;
701:
702:      if cbb74.Text='GruppoB' then
703:      begin
704:      Sorge:=Form2.ABSTable12;
705:      Sorge.Open;
706:      end;
707:      //CONTROLO CATEGORIA
708:      if Edit4.Text='' then
709:      begin
710:      ShowMessage(' Inserisci la Categoria');
711:      Exit;
712:      end;
713:      //Verifica se esiste già l'iscrizione
714:      Sorge.First;
715:      while not Sorge.Eof do
716:      begin
717:      n:=Sorge.FieldByName('CODDSPL').AsInteger;
718:      trovato:=Form2.ABSTable7.Locate(' CODGA;CODDSPL;CODATL',VarArrayOf([Edit1.Text
,IntToStr(n),IntToStr(ATL)]),[]);
719:      if trovato=True then
720:      begin
721:      ShowMessage('ATTENZIONE ! - Questo Atleta risulta già iscritto '
722:      +#13+'ad una o più gare del Gruppo. Non è consentita l''iscrizione automat
ica');
723:      Exit;
724:      end;
725:      Sorge.Next;
726:      end;
727:
728:      //Carica il nuovo record
729:      Sorge.First;
730:      while not Sorge.EOF do // Modo corretto di loop EOF
731:      begin
732:      Form2.ABSTable7.Insert;
733:      Form2.ABSTable7.FieldByName('CODGA').AsInteger:=StrToInt(Edit1.Text);
734:      Form2.ABSTable7.FieldByName('CODDSPL').AsInteger:=Sorge.FieldByName('CODDSPL').
AsInteger;
735:      Form2.ABSTable7.FieldByName('CODATL').AsInteger:= ATL;
736:      Form2.ABSTable7.FieldByName('COGNOME').AsString:=Cogn;
737:      Form2.ABSTable7.FieldByName('NMPET').AsInteger:=0;
738:      Form2.ABSTable7.FieldByName('NMPOS').AsInteger:=0;
739:      Form2.ABSTable7.FieldByName('TAME1').AsDateTime:=StrToTime('00:00:00');
740:      Form2.ABSTable7.FieldByName('TBME2').AsDateTime:=StrToTime('00:00:00');
741:      Form2.ABSTable7.FieldByName('DQ').AsBoolean:=False;
742:      Form2.ABSTable7.FieldByName('PEN').AsDateTime:=StrToTime('00:00:00,000');
743:      Form2.ABSTable7.FieldByName('MANL').AsBoolean:=False;
744:      Form2.ABSTable7.FieldByName('CAT').AsInteger:=StrToInt(Edit4.Text);
745:      Form2.ABSTable7.Post;
746:      Sorge.Next;
747:      end;
748:      Button5.Click;
749:      end; //Iscrizione automatica terminata
750:      end;
751:
752:      //SCHEMA MANUALE
```

```

754:      if (rb2.Checked=True) and (Edit2.Text<>'' ) then      //verifica se già inserito
755:      begin
756:          ds7.DataSet:=ABSQuery7;
757:          with AbsQuery7 do
758:          begin
759:              Close; //cerca i doppioni
760:              SQL.Text := 'Select * from Gara where CODATL = '+ QuotedStr(IntToStr(ATL))+
761:              ' AND CODDSPL = '+QuotedStr(Edit2.Text);
762:              ExecSql;
763:              Open;
764:              volte:=AbsQuery7.RecordCount;
765:              if volte>0 then
766:              begin
767:                  ShowMessage(Nom+' '+Cogn+' è già iscritto '+IntToStr(volte)+' volte a ques
768: ta disciplina');
769:                  Button5.Click;
770:                  Form2.ABSTable2.First;
771:                  Exit;
772:              end;
773:          end;
774:          if Edit4.Text='' then
775:          begin
776:              ShowMessage(' Inserisci la Categoria');
777:              Exit;
778:          end;
779:          if MessageDlg('Confermi l''iscrizione di '+Nom+ ' '+ Cogn,
780: mtconfirmation, [mbYes,mbNo], 0) = mrYes then
781:          begin
782:              Form2.ABSTable7.Insert;
783:              Form2.ABSTable7.FieldName('CODATL').AsInteger:=ATL;
784:              Form2.ABSTable7.FieldName('CODGA').AsInteger:= StrToInt(Edit1.Text);
785:              Form2.ABSTable7.FieldName('CODDSPL').AsInteger:=StrToInt(Edit2.Text);
786:              Form2.ABSTable7.FieldName('COGNOME').AsString:= Cogn;
787:              Form2.ABSTable7.Post;
788:              Button5.Click;          //per ripristinare l'evento si ds72
789:              Form2.ABSTable2.First;
790:              inserisco7:=False;
791:          end else Exit;
792:          end else
793:          begin
794:              if rb2.Checked=True then
795:              begin
796:                  ShowMessage('Iscrizione manuale, indica prima il codice disciplina ');
797:                  Button2.Click;
798:                  exit;
799:              end;
800:          end else
801:          begin
802:              ShowMessage('Inserisci il Codice della Gara. ');
803:              Exit;
804:          end;
805:      end;
806:
807:      //INSERISCE I Tempi(da unit Tempi)
808:      procedure TForm6.dbgrd7CellClick(Column: TColumn);
809:      begin
810:          if inserisco7=True then Exit;
811:
812:          if tempo='' then
813:          begin
814:              ShowMessage('Il valore da inserire è nullo');
815:              Exit;
816:          end;
817:          ABSTable14.Edit;
818:          TDateTimeField(ABSTable14.FieldName('TAME1')).DisplayFormat:='nn:ss,zzz';
819:          TDateTimeField(ABSTable14.FieldName('TBME2')).DisplayFormat:='nn:ss,zzz';
820:          if (rab1 = True) then
821:          begin

```

```

821:         if MessageDlg('Esiste già un dato, vuoi sostituirlo?',
822:             mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
823:         ABSTable14.FieldName('TAME1').AsDateTime:=StrToTime(tempo);
824:     end;
825:     if (rab2 = True) then
826:     begin
827:         if ABSTable14.FieldName('TAME1').AsDateTime<>0 then
828:         if MessageDlg('Esiste già un dato, vuoi sostituirlo?',
829:             mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
830:         ABSTable14.FieldName('TBME2').AsDateTime:=StrToTime(tempo);
831:     end;
832:     ABSTable14.Post;
833:     tempo:='';
834:     //trucco per poter fire l'onclick sulla stessa riga due volte
835:     Form6.dbgrd7.SelectedField:=ABSTable14.FieldName('NM');
836: end;
837:
838:
839: // Ricerca per cognome
840:
841:         // seleziona una lettera dalla combobox per poi inserire l'atleta
842: procedure TForm6.cbb72Select(Sender: TObject);
843: var
844: c:string;
845: begin
846:     if Edit1.Text<>' ' then
847:     begin
848:         c:=cbb72.Items[cbb72.ItemIndex]; //estrae la scelta dalla combobox
849:         scelta:=True; //scelta fatta per cognome
850:         ds72.DataSet:=AbsQuery72;
851:         with AbsQuery72 do
852:         begin
853:             Close; //inserisce una variabile nel query selezionandola per primo carattere
854:             SQL.Text := 'Select * from AnaAtl where substring(COGNOME,1,1) = '+ QuotedStr(c);
855:             ExecSql;
856:             Open;
857:             cbb72.ItemIndex:=-1;
858:         end;
859:     end else
860:     begin
861:         cbb72.ItemIndex:=-1; //riporta la combobox all'inizio
862:         ShowMessage('Inserisci il Codice della Gara');
863:         Exit;
864:     end;
865: end;
866:
867: procedure TForm6.cbb74Change(Sender: TObject); //SCEGLIE IL GRUPPO
868: var //realizza per ogni gruppo scelto una tabella delle DSPL per quella gara
869: dsplSi:Boolean;
870: DSPLn:Integer;
871: baseSi:Boolean;
872: begin
873:     Form2.ABSTable6.Open;
874:     Form2.ABSTable10.TableName:='repBase';
875:     Form2.ABSTable10.Open;
876:     Form2.ABSTable10.Edit;
877:     Form2.ABSTable11.TableName:='repGRPA';
878:     Form2.ABSTable11.Open;
879:     Form2.ABSTable11.Edit;
880:     Form2.ABSTable12.TableName:='repGRPB';
881:     Form2.ABSTable12.Open;
882:     Form2.ABSTable12.Edit;
883:     if Edit1.Text<>' ' then
884:     begin
885:         if cbb74.Text='Base' then // GRUPPO BASE
886:         begin
887:             if Form2.ABSTable6.RecordCount>0 then //controlla se per la Gara scelta l'accoppiame

```

```

888:         begin                                // esiste già
889:             baseSi:= Form2.ABSTable6.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'
esistenza
890:         if baseSi=True then                    //del codice della Gara negli accoppiamenti di
Base
891:             begin
892:                 if MessageDlg('Il Gruppo delle discipline Base di questa Gara'+#13+
893:                     'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo
) = mrYes then
894:                     begin
895:                         ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare
+#13+
896:                             'per rifare gli accoppiamenti Gruppo Base');
897:                         Exit;
898:                     end;
899:                 end else
900:                 begin
901:                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#
13+
902:                         'per definire gli accoppiamenti Gruppo Base');
903:                     Exit;
904:                 end;
905:
906:             end;
907:             Form2.ABSTable10.Close;
908:             Form2.ABSTable10.EmptyTable;
909:             ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo le disciplin
e
910:             with ABSQuery74 do                // del Gruppo Base di quella Gara
911:             begin
912:                 Close; //estrae le discipline relative alla gara ed al Gruppo Base
913:                 SQL.Text := 'Select * from Base where CODGA = '+QuotedStr(edit1.Text);
914:                 ExecSql;
915:                 Open; //la Tabella è formata da tutti i codici disciplina della Base
916:                 First;
917:                 Form2.ABSTable10.Open;
918:                 Form2.ABSTable10.Edit;
919:                 Form2.ABSTable4.Open;
920:                 while(not Eof)do //Crea le tabelle complete delle discipline per quel Gruppo
921:                 begin
922:                     DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della DSPL
923:                     dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod per localizz
are il record di ABS/4
924:                     if dsplSi=True then
925:                         begin
926:                             Form2.ABSTable10.Insert; //Costruisce la tabella completa delle discipl
ine
927:                             Form2.ABSTable10.FieldName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldName
ame('CODDSPL').AsInteger;
928:                             Form2.ABSTable10.FieldName('DESCR').AsString:=Form2.ABSTable4.FieldName
('DESCR').AsString;
929:                             Form2.ABSTable10.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName
e('TMAX').AsDateTime;
930:                             Form2.ABSTable10.Post
931:                         end else
932:                         begin
933:                             ShowMessage('Attenzione ! Manca questo Codice dalla Tabella delle Dis
cipline : '
934:                                 + IntToStr(DSPLn));
935:                             Exit;
936:                         end;
937:                     Next;
938:                 end;
939:
940:             ds74.DataSet:=Form2.ABSTable10;
941:             end;
942:

```



```

944:         ' per la Gara indicata.');
```

945:     end;

946:

947:     if cbb74.Text='GruppoA' then

948:         begin

949:             Form2.ABSTable8.Open;

950:             if Form2.ABSTable8.RecordCount>0 then *//Controlla se per la Gara scelta l'accoppiame*  
*nto di GruppoA*

951:                 begin   *// esiste già*

952:                     baseSi:= Form2.ABSTable8.Locate('CODGA',strToInt(Edit1.Text),[]); *//Verifica l'*  
*esistenza*

953:                     if baseSi=True then                     *//del codice della Gara negli accoppiamenti di Base*

954:                         begin

955:                             if MessageDlg('Il Gruppo A delle discipline di questa Gara'+#13+

956:                                 'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo

957:                             ) = mrYes then

958:                                 begin

959:                                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#13+

960:   'per rifare gli accoppiamenti del Gruppo A');

961:                                     Exit;

962:                             end;

963:                             end else

964:                                 begin

965:                                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#1

966:   3+

967:   'per definire gli accoppiamenti Gruppo A');

968:                                     Exit;

969:                             end;

970:                     Form2.ABSTable11.Close;

971:                     Form2.ABSTable11.EmptyTable;

972:                     ds74.DataSet := ABSQuery74; *//Costruisce una tabella SQL isolando solo le disciplin*  
*e*

973:             with ABSQuery74 do                             *// del Gruppo A di quella Gara*

974:                 begin

975:                     Close; *//estrae le discipline relative alla gara ed al Gruppo A*

976:                     SQL.Text := 'Select \* from GRPA where CODGA = '+QuotedStr(edit1.Text);

977:                     ExecSql;

978:                     Open;   *//la Tabella è formata da tutti i codici disciplina della Base*

979:                     First;

980:                     Form2.ABSTable11.Open;

981:                     Form2.ABSTable11.Edit;

982:                     while(not Eof)do   *//Crea le tabelle complete delle discipline per quel Gruppo*

983:                         begin

984:                             DSPLn:=dbgrd74.Columns[0].Field.AsInteger; *//preleva il cod della DSPL*

985:                             dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); *// usa il cod per localizz*  
*are il record di ABS/4*

986:                             if dsplSi=True then

987:                                 begin

988:                                     Form2.ABSTable11.Insert;         *//Costruisce la tabella completa delle discipl*  
*ine*

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

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

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

992:                                     Form2.ABSTable11.Post

993:                             end else

994:                                 begin

995:                                     ShowMessage('Attenzione ! Manca questo Codice dalla Tabella delle Dis

996:   cipline : '

997:   + IntToStr(DSPLn));

998:                                     Exit;

999:                             end;

```

1000:         end;
1001:
1002:         ds74.DataSet:=Form2.ABSTable11;
1003:     end;
1004:
1005:     ShowMessage('La tabella mostra le discipline del Gruppo A');
1006: end;
1007:
1008:     if cbb74.Text='GruppoB' then
1009: begin
1010:     Form2.ABSTable9.Open;
1011:     if Form2.ABSTable9.RecordCount>0 then //Controlla se per la Gara scelta l'accoppiame
nto di Gruppo B
1012:     begin // esiste già
1013:         baseSi:= Form2.ABSTable9.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'
esistenza
1014:         if baseSi=True then //del codice della Gara negli accoppiamenti di Base
1015:         begin
1016:             if MessageDlg('Il Gruppo B delle discipline di questa Gara'+#13+
1017:                 'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo
) = mrYes then
1018:             begin
1019:                 ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare
'+#13+
1020:                     'per rifare gli accoppiamenti del Gruppo B');
1021:                 Exit ;
1022:             end;
1023:         end else
1024:         begin
1025:             ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#1
3+
1026:                 'per definire gli accoppiamenti del Gruppo B');
1027:             Exit ;
1028:         end;
1029:     end;
1030:     Form2.ABSTable12.Close;
1031:     Form2.ABSTable12.EmptyTable;
1032:     ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo le disciplin
e
1033:     with ABSQuery74 do // del Gruppo B di quella Gara
1034:     begin
1035:         Close; //estrae le discipline relative alla gara ed al Gruppo B
1036:         SQL.Text := 'Select * from GRPB where CODGA = '+QuotedStr(edit1.Text);
1037:         ExecSql;
1038:         Open; //la Tabella è formata da tutti i codici disciplina della Base
1039:         First;
1040:         Form2.ABSTable12.Open;
1041:         Form2.ABSTable12.Edit;
1042:         while(not Eof)do //Crea le tabelle complete delle discipline per quel Gruppo
1043:         begin
1044:             DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della DSPL
1045:             dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod per localizz
are il record di ABS/4
1046:             if dsplSi=True then
1047:             begin
1048:                 Form2.ABSTable12.Insert; //Costruisce la tabella completa delle discipl
ine
1049:                 Form2.ABSTable12.FieldName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldName('CODDSPL').AsInteger;
1050:                 Form2.ABSTable12.FieldName('DESCR').AsString:=Form2.ABSTable4.FieldName('DESCR').AsString;
1051:                 Form2.ABSTable12.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName('TMAX').AsDateTime;
1052:                 Form2.ABSTable12.Post
1053:             end else
1054:             begin
1055:                 ShowMessage('Attenzione ! Mancia questo Codice dalla Tabella delle Dis

```

```
1056:         + IntToStr(DSPLn));
1057:         Exit;
1058:     end;
1059:     Next;
1060: end;
1061:
1062:     ds74.DataSet:=Form2.ABSTable12;
1063: end;
1064:     ShowMessage('La tabella mostra le discipline del Gruppo B');
1065: end;
1066: end else
1067: begin
1068:     ShowMessage('Inserisci il Codice della Gara ');
1069:     cbb74.ItemIndex:=0;
1070:     Exit;
1071: end;;
1072: end; //NAVIGATORE iscrizioni
1073:
1074: procedure TForm6.dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);
1075: begin
1076:     case button of
1077:         nbInsert: //se inserisce un nuovo record
1078:         begin
1079:             if (Edit1.Text='') OR (Edit2.Text='') then
1080:             begin
1081:                 ShowMessage('Inserisci il Codice della Gara e/o della Disciplina');
1082:                 Exit;
1083:             end;
1084:             Form2.ABSTable7.Last;
1085:             Form2.ABSTable7.Append;
1086:             inserisco7:=True; //inserisce il flag nel nuovo record
1087:             dbgrd7.SetFocus;
1088:             DBGrd7.SelectedField := Form2.ABSTable7.FieldName('CODATL');
1089:             end;
1090:             nbPost:
1091:             begin //se lo carica nella tabella
1092:                 inserisco7:=False ;
1093:             end;
1094:             nbDelete: //se cancella il record - serve anche per evitare di entrare in Query
1095:             begin
1096:                 inserisco7:=False;
1097:             end;
1098:             nbEdit:
1099:                 inserisco7:=True;
1100:             end;
1101:         end;
1102:
1103: procedure TForm6.Edit1Change(Sender: TObject);
1104: begin
1105:     if rb1.Checked = True then
1106:         cbb74.ItemIndex:=0;
1107: end;
1108:
1109: end.
1110:
```