

## Costruzione di un report IVA secondo le Norme Vigenti ( 2006 )

Poniamo di dover costruire un report che presenti i seguenti dati e che distingua i CLIENTI dai FORNITORI:

- Dati anagrafici della azienda
- Codice azienda
- Ragione sociale ( 1na o 2 a seconda dell' azienda es. Pippo.snc o Pippo.Snc di Mario Prosperi & C. )
- La somma della parte IMPONIBILE e di quella NON IMPONIBILE
- L' IVA totale sull'imponibile ( in quanto una stessa azienda potrebbe presentarsi più volte in una tabella )
- Il totale dato dall'imponibile sommato all' iva.

Il sistema utilizzato è quello di crearsi in memoria una tabella da cui poi recuperare i dati per fare il report, per inserirli in un file di testo, ecc. ecc. come da FIGURA 1.

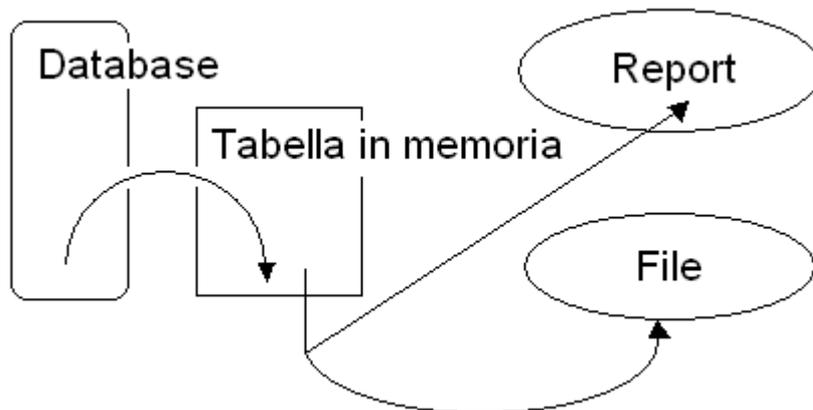


FIGURA 1

Esaminiamo adesso la query che serve a creare la tabella in memoria. La tabella in memoria è un componente che si trova in RXDBAWARE e si chiama RXMEMORYDATA che a sua volta deve essere collegato ad un DATASOURCE. Nel datasource il dataset è il nome della rxmemoridata come nella FIGURA 2.

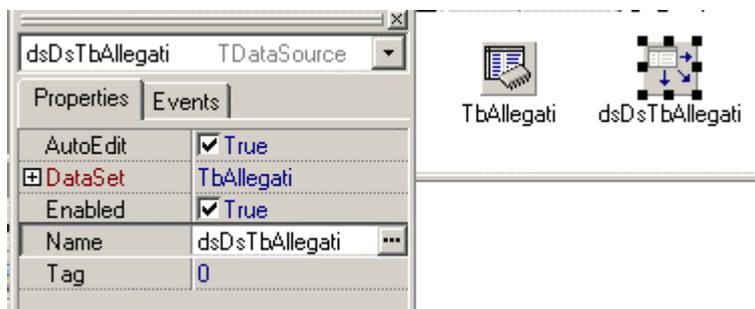


FIGURA 2

Nel FIELD EDITOR nella tballegati che corrisponde all' rxmemorydata, si devono creare tutti i campi che verranno utilizzati nella query, così come si fa in un LOOKUP, così come le abbiamo create in FIGURA 3.

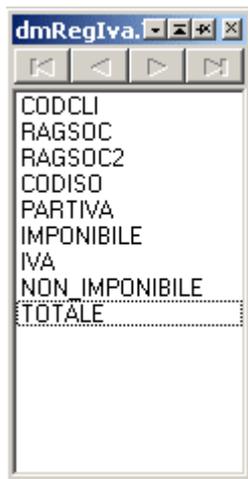


FIGURA 3

La procedura che realizza il report è la procedura comune, cioè che cambia e la routine Apriqry, di cui un tipo potrebbe essere quella evidenziata in giallo, ma che come vedremo è soggetta a restrizioni, in base alla vigente legge.

```

procedure TfmAllegati.BtnPreviewClick(Sender: TObject);
begin
if (apriqry >0) then
begin
//ShowMessage(GlbOption.PathReport);
ppReport1.Template.FileName := GlbOption.PathReport + 'RptAllegati.rtm' ;
ppReport1.Template.LoadFromFile;
ppReport1.Device := dvScreen;
try
Frmpreview := TFrmpreview.Create(Application);
Frmpreview.PreviewReport(ppReport1);
finally
Frmpreview.Free;
end;

end
else
begin
showmessage('Anteprima non disponibile');
end
end;

```

```

function TfmAllegati.Apriqry: Integer;
var
i: Integer;
Vanno : string;
Vanno2 : string;
begin

if selezione = " then
begin
ShowMessage('Devi cliccare su clienti o fornitori');
Result := 0;

```

```
Exit;  
end;
```

```
dmRegIva.TbAllegati.Open;  
dmRegIva.TbAllegati.EmptyTable;
```

```
dmregiva.QryAllegati.Close;  
dmregiva.qryallegati.sql.clear;  
// CLIENTI tipo = c FORNITORI tipo = f
```

```
dmregiva.qryallegati.sql.append('select max(c.rag_sociale) rag_sociale,  
max(c.rag_sociale1) rag_sociale2, max(part_lva) as piva, max(c.cod_iso) as  
iso,max(c.cod_cli) as cod1, sum(R.imponibile) IMPO, sum(R.iva) IVA , T.cod_iva,  
T.aliquota from REGIVA R ');  
dmregiva.qryallegati.sql.append('left outer join CLIFOR C on (R.COD_CLI =  
C.COD_CLI) ');  
dmregiva.qryallegati.sql.append('left outer join TAB_IVA T on (R.cod_iva = T.cod_iva  
)');  
if chk1.Checked = True then //data sola  
//anno in corso  
begin //1  
if selezione = 'clienti' then  
begin //2
```

```
// il singolo anno e un oggetto di 4 caratteri  
dmregiva.qryallegati.sql.append('where R.anno >= ' + edtanno.text + ' and r.anno <=  
' + edtanno.text + ' and c.tipo = "C");  
end //2  
else // fornitori  
begin //3
```

```
// il singolo anno e un oggetto di 4 caratteri  
dmregiva.qryallegati.sql.append('where R.anno >= ' + edtanno.text + ' and r.anno <=  
' + edtanno.text + ' and c.tipo = "F");
```

```
end; //3  
end //1  
else //selezione // 2 date complete non solo dell'anno  
begin //0
```

```
if selezione = 'clienti' then  
begin //2  
Vanno := cambiadata(data1.text,");  
Vanno2 := cambiadata(data2.text,");  
// Non si ricerca più su r.anno ma su r.data che è una data completa  
dmregiva.qryallegati.sql.append('where R.data >= ' + Vanno + ' and r.data <= ' +  
Vanno2 + ' and c.tipo = "C");  
end //2  
else // fornitori  
begin //3  
Vanno := cambiadata(data1.text,");
```

```

Vanno2 := cambiadata(data2.text,"");
dmregiva.qryallegati.sql.append('where R.data >= "' + Vanno + '" and r.data <= "' +
Vanno2 + '" and c.tipo = "F"');
end; //3
end; //0

dmregiva.qryallegati.sql.append('group by r.COD_CLI, T.COD_IVA, t.aliquota ');
dmregiva.qryallegati.sql.append('order by r.COD_CLI,T.COD_IVA, t.aliquota ');

dmRegIva.QryAllegati.Open;

for i:=1 to dmRegIva.QryAllegati.recordcount do
begin

dmRegIva.TbAllegati.Append;
dmRegIva.TbAllegatiCOD_CLI.value := dmRegIva.QryAllegatiCOD_CLI.Value;
dmRegIva.TbAllegatiRAG_SOCIALE.value :=
dmRegIva.QryAllegatiRAG_SOCIALE.Value;
dmRegIva.TbAllegatiCOD_IVA.value :=dmRegIva.QryAllegatiCOD_IVA.Value;

if dmRegIva.QryAllegati.FieldValues['aliquota'] > 0 then
begin
// OPERAZIONE IMPONIBILE
dmRegIva.TbAllegatiIMPONIBILE.Value := dmRegIva.TbAllegatiIMPONIBILE.Value +
dmRegIva.QryAllegatiIMPO.Value;
dmRegIva.TbAllegatiIVA.Value := dmRegIva.TbAllegatiIVA.Value +
dmRegIva.QryAllegatiIVA.Value;
end
else begin
// OPERAZIONE NON IMPONIBILE
dmRegIva.TbAllegatiNON_IMPONIBILE.Value :=
dmRegIva.TbAllegatiNON_IMPONIBILE.Value + dmRegIva.QryAllegatiIMPO.Value;
end;
dmRegIva.TbAllegati.Post;

dmRegIva.QryAllegati.Next;
end;
dmRegIva.TbAllegati.First;
for i:=1 to dmRegIva.TbAllegati.recordcount do
begin
dmRegIva.TbAllegati.next;
end;
dmRegIva.TbAllegati.First;
result := dmRegIva.QryAllegati.recordcount;
end;

```

In questo caso la query degli allegati che a sua volta è collegata ad un datasource ( FIGURA 4 ) e che ha la lista dei campi da dover utilizzare ( FIGURA 5 ) :



FIGURA 4

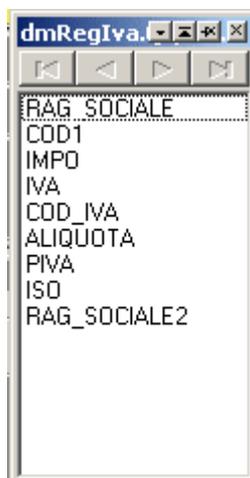


FIGURA 5

Ecco infine la query da inserire in QryAllegati:

```
dmregiva.qryallegati.sql.append('select max(c.rag_sociale) rag_sociale,
max(c.rag_sociale1) rag_sociale2, max(part_iva) as piva, max(c.cod_iso) as
iso,max(c.cod_cli) as cod1, sum(R.imponibile) IMPO, sum(R.iva) IVA , T.cod_iva,
T.aliquota from REGIVA R ');
```

```
dmregiva.qryallegati.sql.append('left outer join CLIFOR C on (R.COD_CLI =
C.COD_CLI) ');
```

```
dmregiva.qryallegati.sql.append('left outer join TAB_IVA T on (R.cod_iva = T.cod_iva)
');
```

```
dmregiva.qryallegati.sql.append('where R.anno = :anno ');
```

```
dmregiva.qryallegati.sql.append('group by r.COD_CLI, T.COD_IVA, t.aliquota ');
```

```
dmregiva.qryallegati.sql.append('order by r.COD_CLI, T.COD_IVA, t.aliquota ');
```

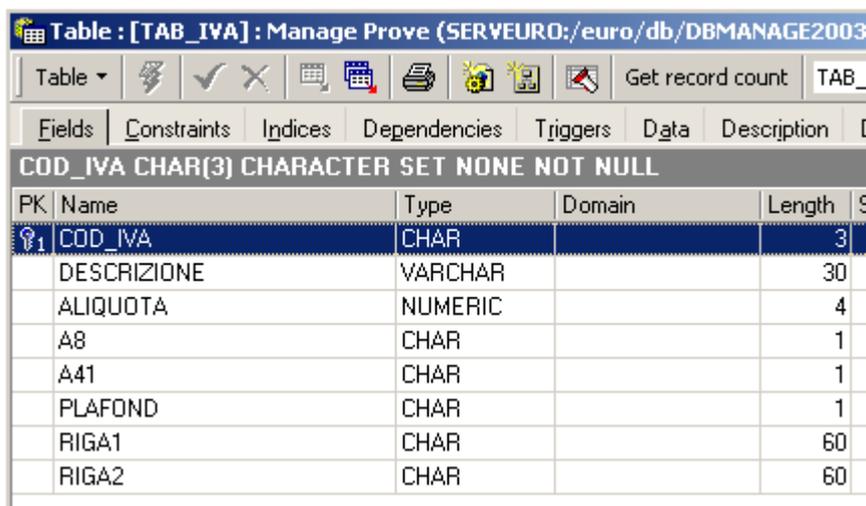
```
dmRegIva.QryAllegati.Open;
```

Invece di utilizzare add per aggiungere righe alla query si utilizza append per poterlo fare su più righe.

```
select max(c.rag_sociale) rag_sociale, max(c.rag_sociale1) rag_sociale2,
max(part_iva) as piva, max(c.cod_iso) as iso,max(c.cod_cli) as cod1,
sum(R.imponibile) IMPO, sum(R.iva) IVA , T.cod_iva, T.aliquota from REGIVA R
left outer join CLIFOR C on (R.COD_CLI = C.COD_CLI) left outer join TAB_IVA T on
(R.cod_iva = T.cod_iva) where R.anno = :anno group by r.COD_CLI, T.COD_IVA,
t.aliquota order by r.COD_CLI, T.COD_IVA, t.aliquota
```

La query in pratica afferma : seleziona il massimo dalla tabella CLIENTI/FORNITORI CLIFOR di ragione sociale e ragione sociale 2 ( *rag\_sociale*, *rag\_sociale1* ) , la partita iva , il cod\_ISO ( CODICE PER L' IVA A LIVELLO EUROPEO ) e il codice cliente, sommando l'imponibile ( da IVA *part\_iva* ) tutto in IMPO e tutta l' IVA in IVA, estraendone il codice iva e l'aliquota dalla tabella REGIVA, il tutto collegando le 2 tabelle REGIVA R e CLIFOR C e TABELLA IVA T in cui il codice dell' iva fra regiva e tabiva sia uguale, dove l'anno di REGIVA sia l'anno passato come parametro, raggruppati per codice cliente di regiva, codice iva di tabiva e ordinati in base al codice cliente di regiva, al codice iva e aliquota di tab\_iva.

Detto in parole povere noi abbiamo 3 tabelle, la tabella clienti/fornitori, la tabella di registrazione dell' iva REGIVA e la tabella del tipo di iva TAB\_IVA.



PK	Name	Type	Domain	Length	S
1	COD_IVA	CHAR		3	
	DESCRIZIONE	VARCHAR		30	
	ALIQUOTA	NUMERIC		4	
	A8	CHAR		1	
	A41	CHAR		1	
	PLAFOND	CHAR		1	
	RIGA1	CHAR		60	
	RIGA2	CHAR		60	

TAB\_IVA

La tabella TAB\_IVA descrive per ogni codice iva il tipo di iva applicato con la rispettiva aliquota. Ecco di seguito un esempio dei dati presenti nella tabella :

COD_IVA	DESCRIZIONE	ALIQUOTA	A8	A41	PLAFOND	RIGA1
04	ALIQUOTA IVA 4%		4,00	N	N	<null>
10	ALIQUOTA IVA 10%		10,00	N	N	<null>
13	ALIQUOTA IVA 13%		13,00	N	N	<null>
20	ALIQUOTA 20%		20,00	N	N	<null>
20I	ALIQU. 20% IMPORTAZIONE		20,00	N	N	<null>
20R	ALIQUOTA IVA 20% REV.CHARGE		20,00	N	N	<null>
20S	FORNITURA REP. SAN MARINO		20,00	N	N	<null>
4*	ALIQUOTA IVA 4%		4,00	N	N	<null>
A07	F.c. IVA art. 7 dpr 633/72		0,00	N	N	<null>
A08	Esente IVA Art. 8 Comma 2		0,00	S	N	<null>
A10	ESENTE ART. 10		0,00	N	N	<null>
A15	ESENTE ART. 15		0,00	N	N	<null>
A40	ESENTE IVA ART. 40 (Vend. UE)		0,00	N	N	Non Im
A41	Esente Art. 41		0,00	N	S	<null>
E26	ESC. IVA ART. 26	<null>		N	N	ESCLU
FC	FUORI CAMPO IVA		0,00	N	N	<null>
N19	NON IMP.LE ART. 9		0,00	N	N	<null>

TAB\_IVA

In questo caso ci sono aliquote del 4%, del 10%, ecc. ecc., e aliquote esenti....

Eccoci adesso ad applicare le restrizioni legislative e a creare il nuovo form.

- Per prima cosa si deve stabilire se si vuole solo l'anno in corso o una diversa disposizione temporale nel report.

- A questo bisogna aggiungere che la legge prevede una differenziazione fra il report dei clienti e quello dei fornitori.

Ecco allora come si dovrebbe presentare all' inizio la routine apriqry....successivamente il form in fase di progettazione e in running per comprendere meglio il pezzo di software qua sotto. ( FIGURA 6,7,8).

```

if chk1.Checked = True then //data sola
//anno in corso
begin //1
if selezione = 'clienti' then
begin //2

// il singolo anno e un oggetto di 4 caratteri
dmregiva.qryallegati.sql.append('where R.anno >= ' + edtanno.text + ' and r.anno <= ' +
edtanno.text + ' and c.tipo = "C"');
end //2
else // fornitori
begin //3

// il singolo anno e un oggetto di 4 caratteri
dmregiva.qryallegati.sql.append('where R.anno >= ' + edtanno.text + ' and r.anno <= ' +
edtanno.text + ' and c.tipo = "F"');

end; //3
end //1
else //selezione // 2 date complete non solo dell'anno
begin //0

if selezione = 'clienti' then
begin //2
Vanno := cambiadata(data1.text,"");
Vanno2 := cambiadata(data2.text,"");
// Non si ricerca più su r.anno ma su r.data che è una data completa
dmregiva.qryallegati.sql.append('where R.data >= ' + Vanno + ' and r.data <= ' +
Vanno2 + ' and c.tipo = "C"');
end //2
else // fornitori
begin //3
Vanno := cambiadata(data1.text,"");
Vanno2 := cambiadata(data2.text,"");
dmregiva.qryallegati.sql.append('where R.data >= ' + Vanno + ' and r.data <= ' +
Vanno2 + ' and c.tipo = "F"');
end; //3
end; //0

```

Come si vede dalle parti evidenziate in giallo se si ha a che fare con una data formata dal solo anno si controlla il campo ANNO, altrimenti se si ha a che fare con date complete si controlla il campo data

Nel trattare il secondo caso però dobbiamo annullare un errore di calcolo della data, che dipende dal fatto che la data non è costituita solo dall'anno ma anche dal mese e dal giorno, che all' interno del database sono salvati come mese, giorno, anno e non come giorno, mese, anno ;successiva figura 7 ( i cambiamenti come in sezione rossa ):

```
Vanno := edtanno.text;
Vanno2 := edtanno.text;

Vanno := cambiadata(data1.text,");
Vanno2 := cambiadata(data2.text,");
```

```
function TfmAllegati.cambiadata(DataN : string ;Locazione: string):string;
var
lung:Integer;
//lung2:Integer;
Carattere:string;
stringa :string;
//stringa2:string;
//nuovadata :string;
a:Integer;
//b:Integer;
//contatore:Integer;

begin //1
if (locazione = "") then
begin //1b
lung := Length(dataN);
for a:=1 to lung do
begin //2
Carattere := leftstr(dataN,1);
//ShowMessage( 'CARATTERE = ' + carattere);
if (Carattere <> '/') then
begin //3
//Contatore := contatore + 1;
stringa := stringa + Carattere; // giorno
//ShowMessage('STRINGA=' + stringa);
DataN := sTRuTILs.rightStr(datan, (lung - A));
end //3
else
begin //4
DataN := sTRuTILs.rightStr(datan, (lung - (A)));
//ShowMessage( 'DATAN = ' + DATAN);
Result := cambiadata2(datan,stringa); // datan = 12/2006 - stringa = 10 (10/12/2006)
end //4
end; //2
end //1
//end;
else
begin //1b
if (Locazione = 'i') then
begin
```

```

Result := '1/1/' + DataN;
end
else
begin
Result := '12/31/' + DataN;
end;
end; //1b
end;

```

```

function TfmAllegati.cambiadata2(DataN : string; stringa :string):string;
var
lung2:Integer;
Carattere2:string;
stringa2:string;
//nuovadata2 :string;
b:Integer;
//contatore2:Integer;

begin //1
lung2 := Length(dataN);
for b:=1 to lung2 do
begin //2
Carattere2 := leftstr(dataN,1);
//ShowMessage('CARATTERE = ' + carattere2);
if (Carattere2 <> '/') then
begin //3
//Contatore2 := contatore2 + 1;
stringa2 := stringa2 + Carattere2; // giorno
//ShowMessage(stringa2);
DataN := sTRuTILs.rightStr(datan, (lung2 - B));
end //3
else
begin //4
DataN := sTRuTILs.rightStr(datan, (lung2 - (B)));
//ShowMessage( 'DATAN = ' + DATAN);
Result:= stringa2 + '/' + stringa + '/' + DATAN ; // datan = 2006 - stringa2 = 12
(10/12/2006)
end //4
end; //2
end; //1

```



FIGURA 6

Il nuovo form prevede che si selezioni fra fornitori e clienti, che si scelga l'anno in corso ( 5 checkbox 5 e casella di testo 6 ) o altrimenti che si possa inserire un lasso di tempo in data ( questa parte non è visibile se non si deseleziona il checkbox 5. Ecco come il form si presenta all' avvio ( FIGURA 7 ) e come si presenta con la scelta del lasso di tempo nella query di ricerca ( FIGURA 8 ). Da notare che per costruire la query si utilizza [dmregiva.qryallegati.sql.append](#).



FIGURA 7

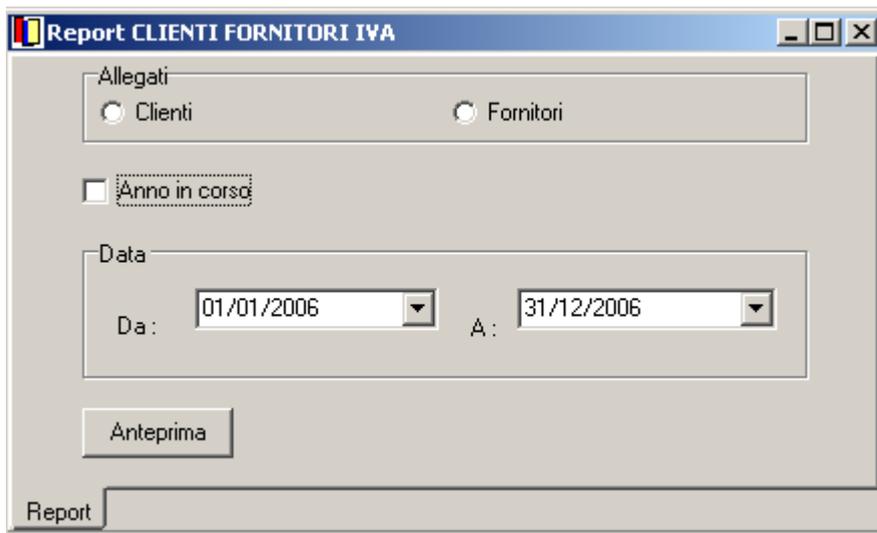


FIGURA 8

Una volta cliccato su anteprima ed eseguito il codice precedente si apre la query:

```
dmRegIva.QryAllegati.Open;
```

Per richiamare i record trovati si usa:

```
procedure TfmAllegati.BtnPreviewClick(Sender: TObject);
begin
if (apriqry > 0) then // cambiata da > a =
  begin
  //ShowMessage(GlbOption.PathReport);
  ppReport1.Template.FileName := GlbOption.PathReport + 'RptAllegati.rtm' ;
  ppReport1.Template.LoadFromFile;
  ppReport1.Device := dvScreen;
  try
  Frmpreview := TFrmpreview.Create(Application);
  Frmpreview.PreviewReport(ppReport1);
  finally
  Frmpreview.Free;
  end;

  end
else
begin
  showmessage('Anteprima non disponibile');
end
end;
```

Che in pratica controlla se il numero dei record ottenuti è maggiore di 0, apre il report e lo manda a schermo o invia un messaggio per avvertire che l'anteprima non è disponibile.

Utilizzando il metodo LOOKUP si aggiungono i campi alla tabella TBAlegati e alla Query QRYallegati, in particolare vedremo come si aggiunge il Codice Cliente COD\_CLI ( FIGURA 9 ).

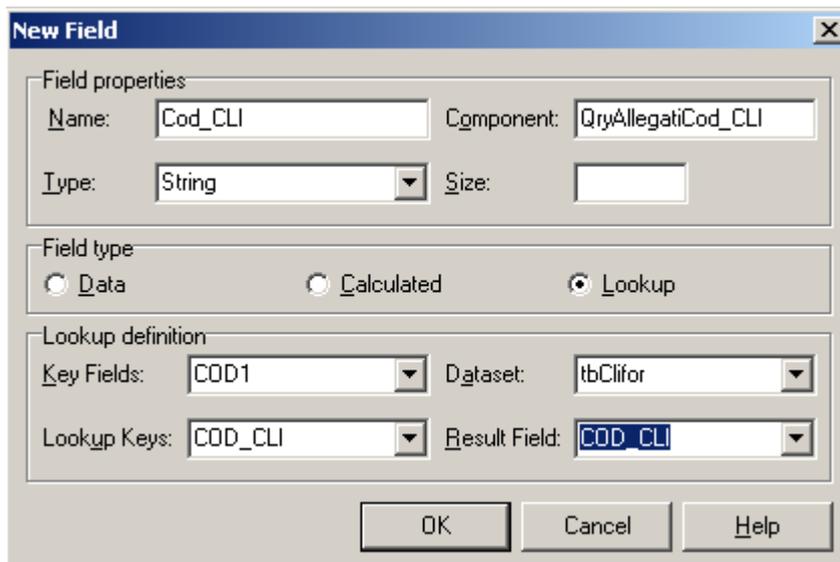


FIGURA 9

Così dovrebbero apparire la tabella TAllegati ( FIGURA 10 ) con i suoi campi e la query QRYallegati ( FIGURA 11 ).

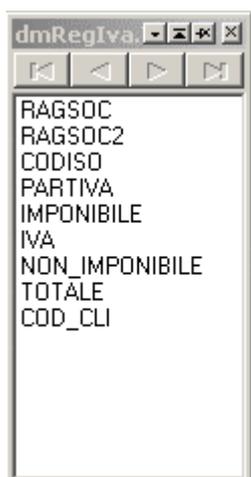


FIGURA 10

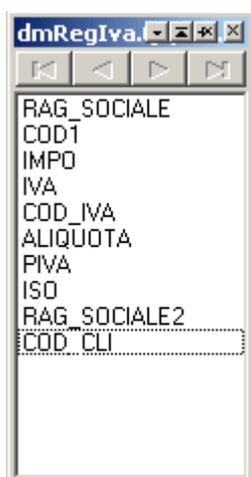


FIGURA 11

Passiamo adesso alla creazione del vero e proprio report.  
Per prima cosa aggiungiamo un elemento per visualizzare ad esempio la data.

Per far ciò, dato che il valore viene visualizzato solo una volta all'apertura del report non creeremo un nuovo field, ma ci basterà una variabile ( di seguito introdurrò anche come creare un nuovo field.

- 1) si inserisce una variabile a cui si assegna un tipo all'interno del report ( FIGURA 12 )
- 2) si crea una variabile globale di modulo dove salvare il valore da assegnare a tale variabile
- 3) Si clicca sulla variabile creata nel report e fra gli eventi si sceglie oncalc
- 4) Doppio clic su oncalc e alla routine che appare si associa il valore globale alla variabile del report.
- 5) Nel codice del programma si inserisce un valore per la variabile.....

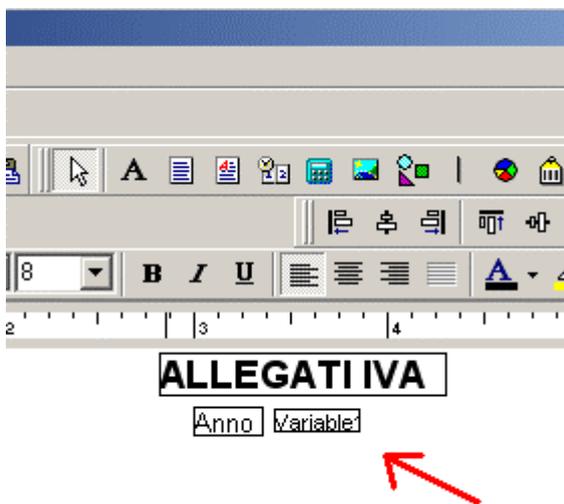


FIGURA 12

```
var
  fmAllegati: TfmAllegati;
  TipoQry:integer;
  cambiotesto1:Boolean;
  cambiotesto2:Boolean;
  cambiodata1:Boolean;
  cambiodata2:Boolean;
  selezione:string; // radio buttons
  ANNO : string;
```

```
procedure TfmAllegati.ppVariable1Calc(Sender: TObject; var Value: Variant);
begin
  ppVariable1.Value:= ANNO;
end;
```

---

Il secondo metodo invece parte dal presupposto che si abbia il bisogno di creare un campo di un record ed io lo metto qui solo come informazione se volete lo potete saltare. Bisogna in pratica creare nuovi campi all'interno della tabella in memoria.

Su ppdbpipeline1 si posta AUTOCREATEFIELDS a true Poi nella tabella in memoria si deve selezionare active a false. A questo punto, sempre sulla tabella in memoria, in questo caso la tabella allegati, si seleziona dal menu a discesa FIELDS EDITOR – Dall'interno di fields editor si clicca quindi sul menu a discesa NEW FIELD e si aggiunge un campo di tipo data... ( FIGURA 13 ).

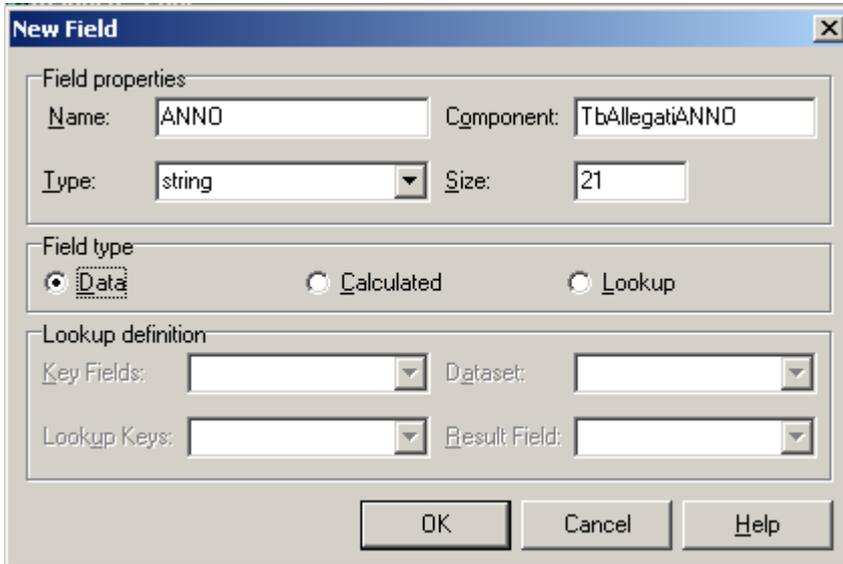


FIGURA 13

Ciò che si ottiene è un nuovo campo fra quelli della tabella ( FIGURA 14 ) :

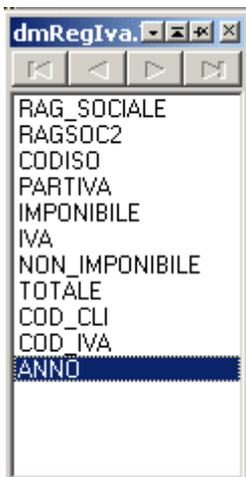


FIGURA 14

Campi del nostro progetto come IMPONIBILE e come NON\_IMPONIBILE sono creati in questa maniera, definendoli come DATA.

Dopo aver aggiunto i valori fissi si passa alla ulteriore costruzione delle parti del report. Si aggiunge il numero di pagina ( FIGURA 15 ) e le varie label dell'intestazione ( FIGURA 16 ). Dopo l'intestazione si aggiungono i campi relativi ai fields della tabella che dovranno essere stampati, a video o verso la stampante. Per poter aggiungere i fields si utilizzano i DB TEXT che faranno riferimento alla DBpipeline ( FIGURA 17 ). Ad esempio nella immagine precedente si aggiunge il campo RAGIONE SOCIALE alla tabella. All'interno della pagina si possono anche inserire linee di divisione, immagini ed altro... ( FIGURA 18

). Inoltre è anche possibile effettuare delle operazioni come la somma dei campi ( FIGURA 19 ).

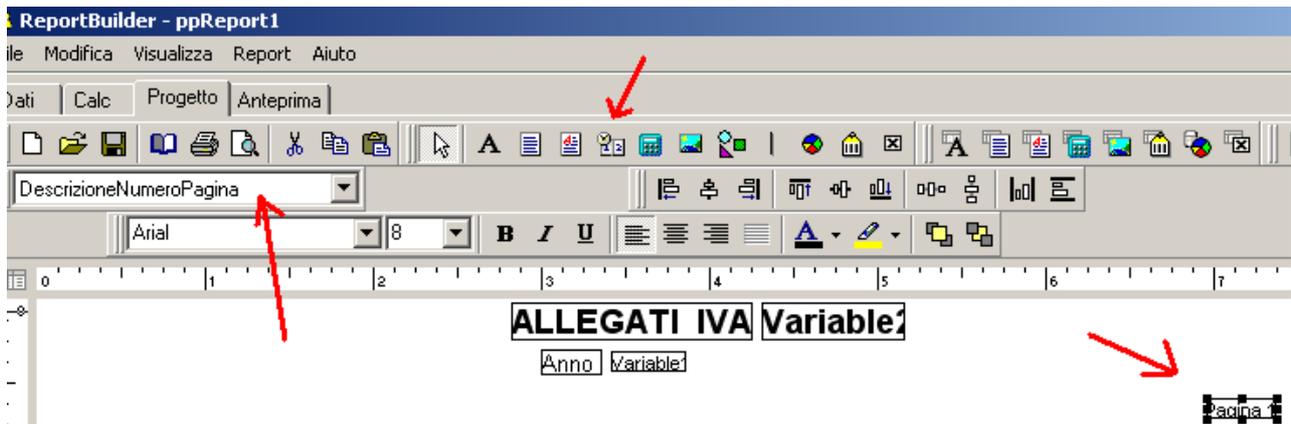


FIGURA 15

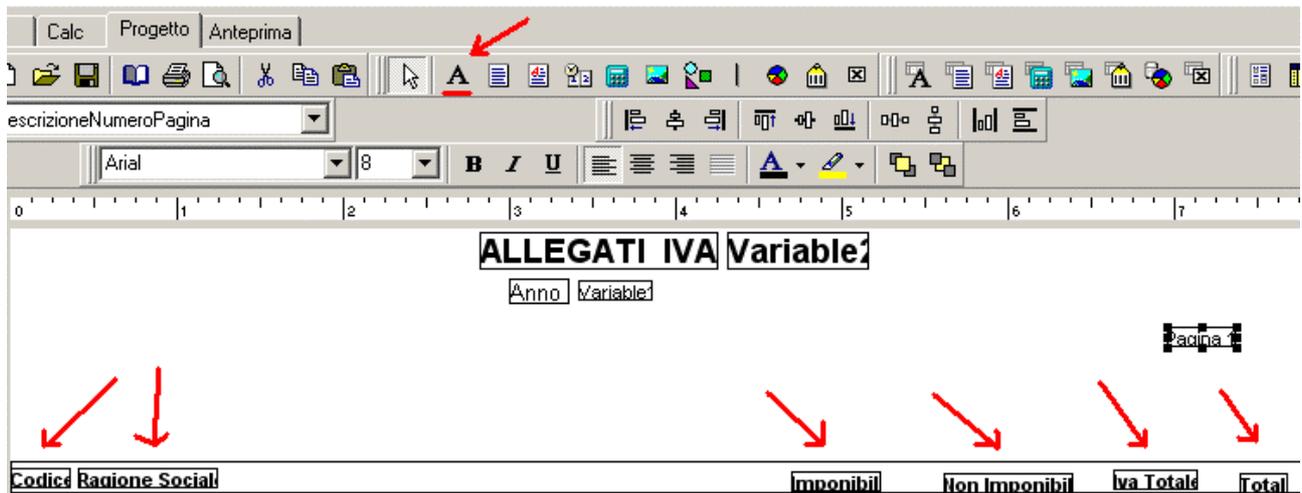


FIGURA 16

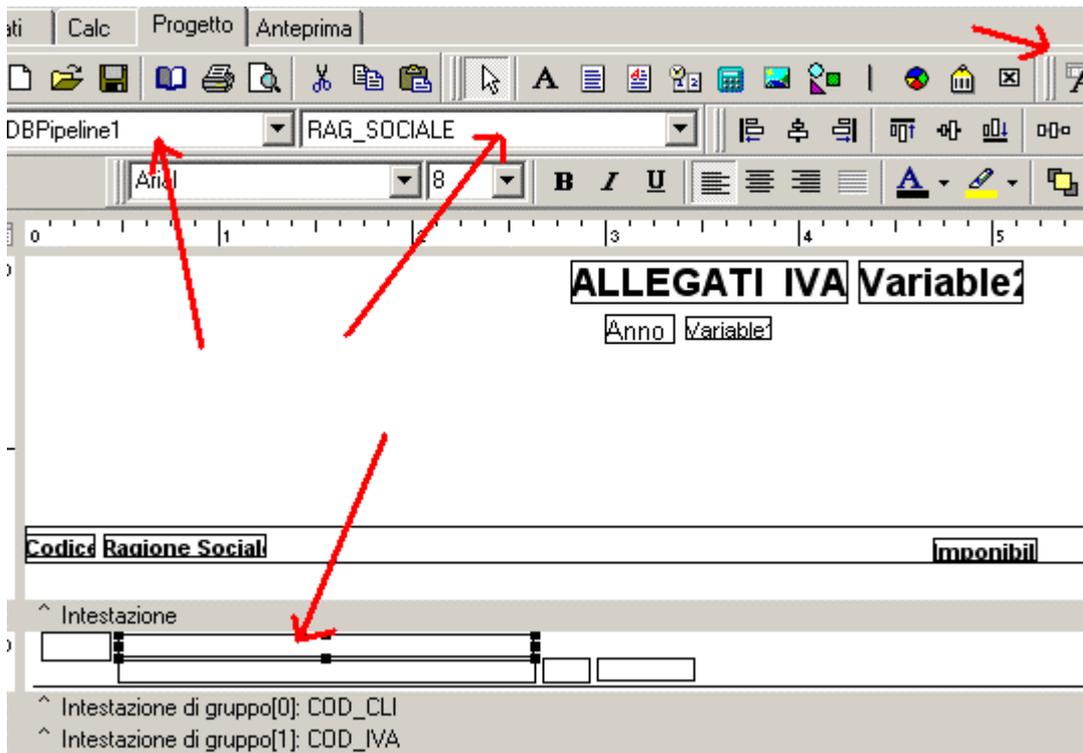


FIGURA 17

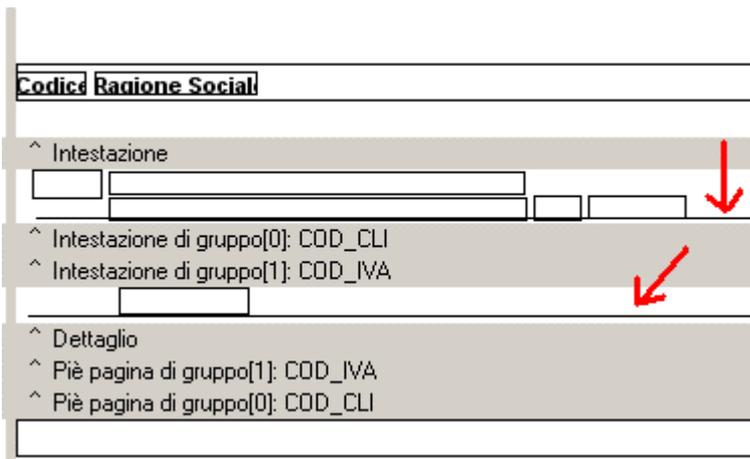


FIGURA 18

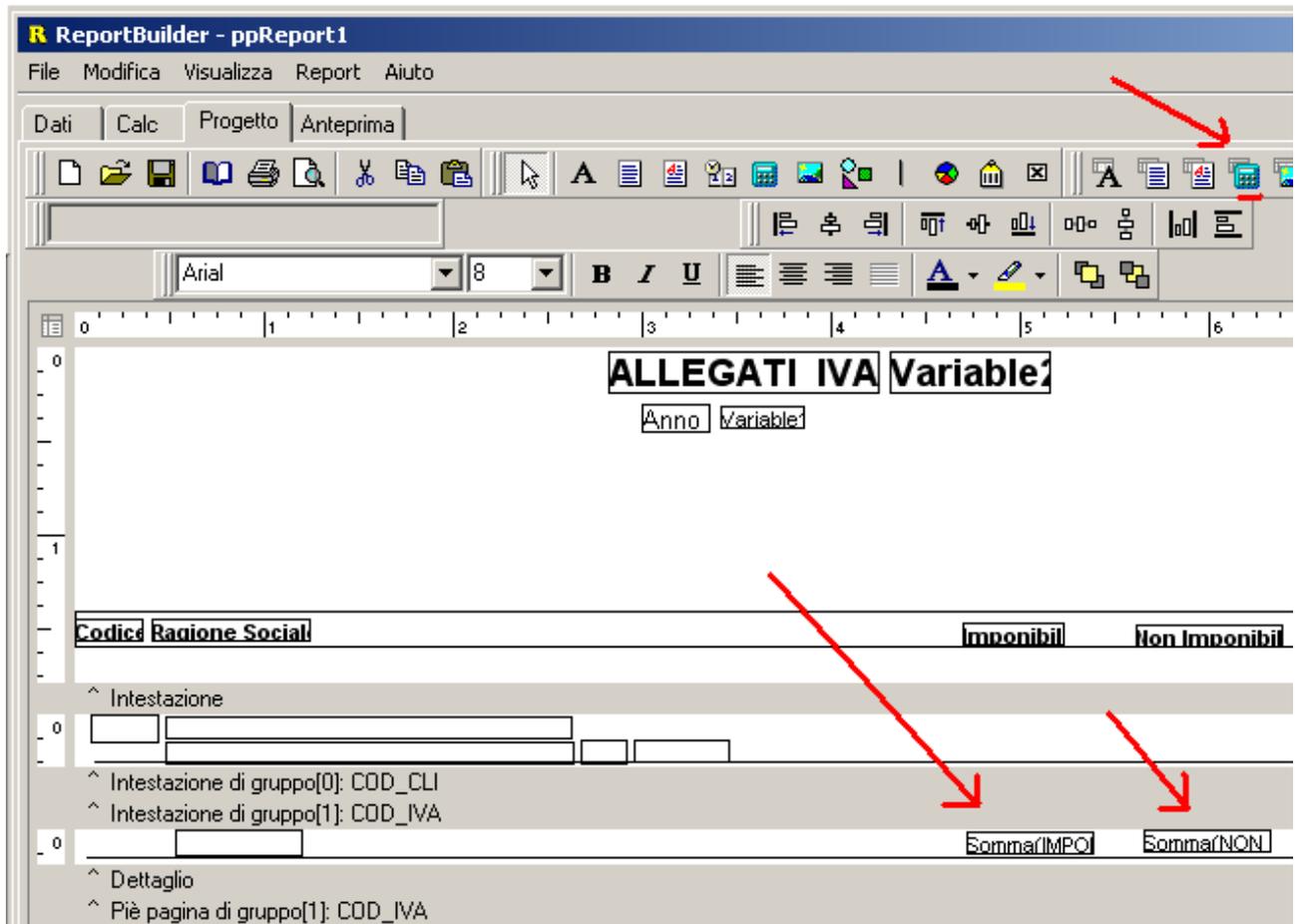


FIGURA 19