

## FARE UN REPORT CON IL COMPONENTE PPREPORT

e problemi connessi

Iniziamo la nostra discussione ponendo un caso, riguardante un database, da cui vogliamo estrarre un certo numero di dati e presentarli in un certo modo:

Le tabelle del database di cui non ci interessa altro, presentano i seguenti campi.... Nella tabella CLIENTI/FORNITORI che chiameremo CLIFOR dovremo separatamente, prima per i clienti e poi per i fornitori estrarre il totale imponibile di tutte le operazioni, il totale iva idem ed infine il totale del documento es...

CLIENTE X	TOTIMP1	TOTIVA1	
CLIENTE X	TOTIMP2	TOTIVA2	
CLIENTE X	TOTIMP3	TOTIVA3	
			TOTALE DOCUMENTO

Se nel database i campi sono in questo modo è ovvio che il report non dovrà presentare ogni singola operazione ma solo i totali per cliente o per fornitore dell'imponibile edell' iva. Tralasciamo per il momento gli altri campi del database come ad esempio il tipo di operazione Acquisto, Vendita, Corrispettivi..... o i codici dei clienti/fornitori.....

Ecco come si presenterebbe la SQL per fare l'operazione spiegata di seguito....

```
select C.cod_cli, C.rag_sociale,R.tot_documento, R.imponibile, R.iva  
from CLIFOR C left outer join REGIVA R on (R.COD_CLI = C.COD_CLI)  
where (C.COD_CLI >=:c1) and (C.COD_CLI<=:c2) order by C.COD_CLI, R.ANNO,  
R.TIPO
```

Partiamo dalle 2 tabelle che ci interessano e cioè CLIFOR e REGIVA che chiameremo C e R . la query si presenta dunque in questo senso :

**seleziona dalla tabella clienti/fornitori il codice cliente e la ragione sociale, poi dalla tabella registrazione iva l'imponibile e l'iva facendo in modo che le due tabelle si colleghino fra loro ( tenendo conto che outer join include anche i record non corrispondenti trovati della tabella di sinistra left ) in base al codice cliente delle due tabelle dove il codice cliente della tabella fornitori sia entro un range rappresentato da C1 e C2, ordinando poi il tutto in base al codice cliente, all' anno e al tipo nella tabella Clienti/Fornitori.**

Andiamo adesso su Delphi e apriamo un form e un datamodule nel form (FIGURA 1 ) inseriremo un pulsante e i 2 componenti utili a fare il PP report : PPREPORT (C), e PPDBpipeline (B).

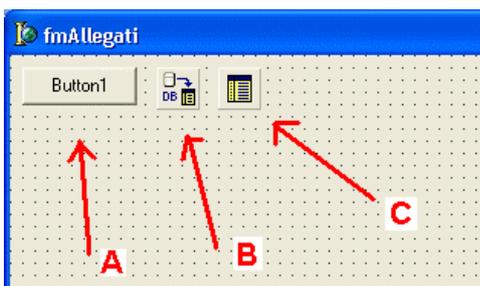


FIGURA 1

Il PPREPORT è ovvio servirà a fare un MODELLO del report che come si vedrà sarà salvato su un file a parte, mentre il PPDBPIPELINE serve a collegare il report al database e cioè agli elementi che vedremo ora nel dataset ( FIGURA 2).



FIGURA 2

Nel DataSource il DataSet corrisponde alla query, nel database l' alias name e il database name sono riempiti al solito con i campi corrispondenti e in più i parametri login e password sono al loro posto, come al solito.

La query (E) ha alla voce SQL la stringa che abbiamo visto prima e alla voce PARAMS i 2 parametri che devono essere riempiti e che fanno parte della sql stessa ( IMMAGINE 3 ).

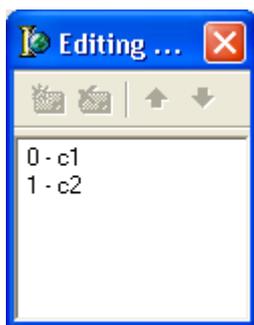


IMMAGINE 3

Il PPDBpipeline è collegato attraverso la voce DATASOURCE al corrispettivo datasource del datamodule ed è così che si ottiene il collegamento al database....

Invece il PPReport1 (C) si collega al ppdbpipeline alla voce DATAPIPELINE.

Alla procedura create del form viene inserita una istruzione per massimizzare il report:

```
procedure TfmAllegati.ppReport1PreviewFormCreate(Sender: TObject);
begin
ppReport1.PreviewForm.WindowState:=wsMaximized;
end;
```

Nel formactivate della stessa form invece si ricollega il datasource al report perché talvolta quest'ultimo non viene preso in considerazione se non è inserito anche a runtime, forse un bug di Delphi.

```
procedure TfmAllegati.FormActivate(Sender: TObject);
begin
ppDBPipeline1.DataSource := dmAllegati.DataSource1;
end;
```

La procedura che riguarda il pulsante invece serve a dare il device per il report, cioè dove verrà visualizzato, e l'avvio della presentazione:

```
procedure TfmAllegati.Button1Click(Sender: TObject);
begin
if apriqry>0 then
  begin

    ppReport1.device:=dvScreen;
    ppReport1.print;
  end
else
  showmessage('Anteprima non disponibile');
end;
```

A quest'ultimo sarà collegata anche una funzione per l'inserimento dei dati mancanti nella query e per aprire quest'ultima. In pratica si passeranno semplicemente i parametri riguardanti i codici dei clienti, in questo caso si dirà alla query di passare tutti i record fra nulla e zzzz, in pratica tutti.....

```
function TfmAllegati.Apriqry: Integer;
begin
  dmAllegati.QryAllegati.close;
  dmAllegati.QryAllegati.ParamByName('c1').value := '';
  dmAllegati.QryAllegati.ParamByName('c2').value := 'zzzz';
  dmAllegati.QryAllegati.Open;
  result:=dmAllegati.QryAllegati.RecordCount;
end;
```

Ora passeremo alla parte più interessante che riguarda il PPREPORT, ovvero la sua modalità di costruzione... Per prima cosa posizionandoci con il mouse sul componente e premendo il tasto destro del mouse potremo selezionare la voce REPORT DESIGNER ecco cosa ci apparirà..... ( con alcune nostre aggiunte che spiegheremo di seguito ) ( IMMAGINE 4 )

La prima cosa da fare è collegare il PPreport ad un database dove estrarre i vari campi che serviranno in seguito al PPreport stesso. Ad esempio facciamo il caso di avere il seguente datamodule di cui utilizzeremo per il report i componenti indicati dalle frecce rosse, datasource2 e query2 ( IMMAGINE 4B ).

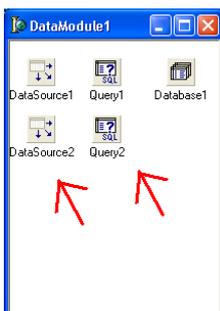


IMMAGINE 4B

Cosa molto importante da ricordarsi è di riempire i campi della query del datamodule sennò in un secondo momento non appariranno fra quelli del report... ( IMMAGINE 4C ).  
 Su query2 tasto destro del mouse selezionare fields editor e poi di nuovo con il tasto destro add fields. Ciò che si otterrà è rappresentato dalla figura 4C. I fields sono stati tutti aggiunti alla query.....

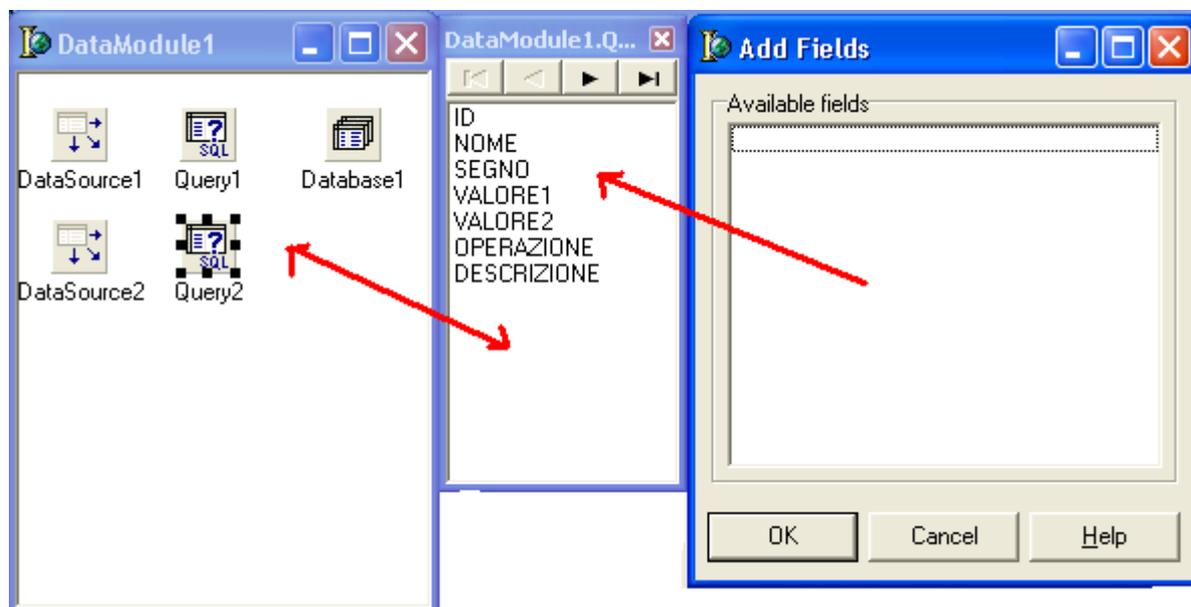


IMMAGINE 4C

In questo modo tutti i campi potranno apparire all'interno del report...

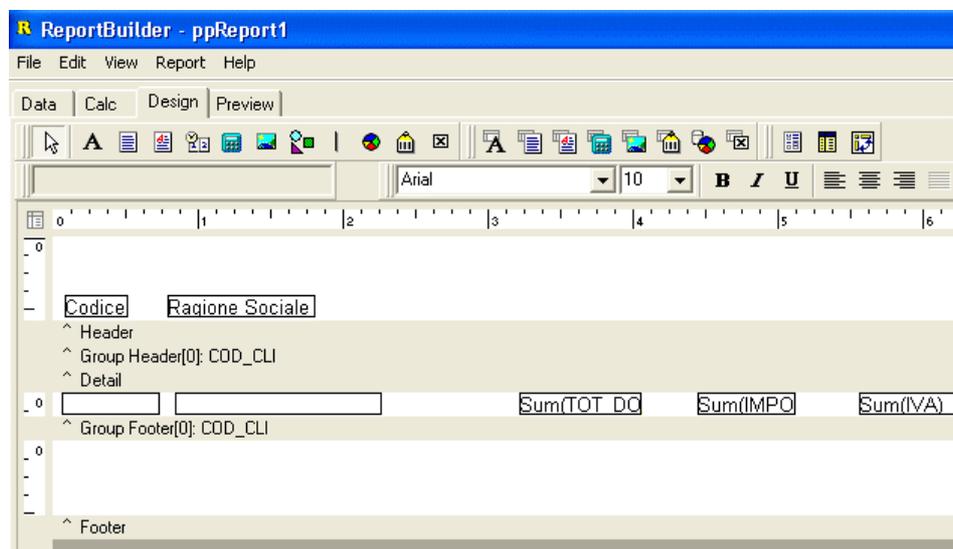


IMMAGINE 4

Utilizzando i componenti che si trovano nel report builder si inseriranno le label per identificare i campi e le caselle di testo per inserirvi i dati. Dato importante qui vi è la caratteristica peculiare di poter aggiungere somme di valori ( in basso a destra ) tali da dover a noi evitare un modo di farlo nel documento....

Complichiamo un poco il programma solo per far vedere come si potrebbe presentare in maniera un po' più complessa e completa ( IMMAGINE 5 ) :



IMMAGINE 5

Ed ecco il nuovo codice da inserire per poter utilizzare i 2 combobox inseriti adesso nella form...( ATTENZIONE AI DATI IN QUANTO E' QUI CHE HO FATTO L' ERRORE di inserire al posto di F100 il valore F01 ).

CODICE ERRATO :

```
function TfmAllegati.Apriqry: Integer;
begin
    dmAllegati.QryAllegati.close;
    if cbda.ItemIndex > -1 then
        begin
            dmAllegati.QryAllegati.ParamByName('c1').value := cbda.Items[cbda.ItemIndex];
        end
    else
        begin
            dmAllegati.QryAllegati.ParamByName('c1').value := cfda.text;
        end;
    if cba.ItemIndex > -1 then
        begin
            dmAllegati.QryAllegati.ParamByName('c2').value := cba.Items[cba.ItemIndex];
        end
    else
        begin
            dmAllegati.QryAllegati.ParamByName('c2').value := cfa.text;
        end;

    dmAllegati.QryAllegati.Open;
    result:=dmAllegati.QryAllegati.RecordCount;
end;

procedure TfmAllegati.aggiungicodici();
var
f:integer;
begin
for f:=1 to 999 do
begin
CBda.items.add('F0' + inttostr(f));
end;
for f:=1 to 999 do
begin
```

```

CBda.items.add('C0' + inttostr(f));
end;
for f:=1 to 999 do
begin
CBA.items.add('F0' + inttostr(f));
end;
for f:=1 to 999 do
begin
CBA.items.add('C0' + inttostr(f));
end;
end;

```

```

procedure TfmAllegati.FormCreate(Sender: TObject);
begin
aggiungicodici();
end;

```

CODICE GIUSTO:

```

procedure TfmAllegati.aggiungicodici();
var
f:integer;
begin
for f:=100 to 999 do
begin
CBda.items.add('F0' + inttostr(f));
end;
for f:=100 to 999 do
begin
CBda.items.add('C0' + inttostr(f));
end;
for f:=100 to 999 do
begin
CBA.items.add('F0' + inttostr(f));
end;
for f:=100 to 999 do
begin
CBA.items.add('C0' + inttostr(f));
end;
for f:=100 to 999 do
begin
LBC.items.add('F0' + inttostr(f));
end;
for f:=100 to 999 do
begin
LBC.items.add('C0' + inttostr(f));
end;
lbc.enabled := false;
rb.Checked := false;
end;

```

```
procedure TfmAllegati.FormCreate(Sender: TObject);
begin
aggiungicodici();
end;
```

la prima cosa da fare è determinare se nelle 2 combobox è stato selezionato qualcosa, poi una altra routine si incarica di riempire i CB dei codici utilizzati per selezionare i Clienti/Fornitori che verrà richiamata alla creazione del form ( terza routine ).

Il problema che mi dava il codice errato è cioè che apparivano nel report solo in dati del database che sono distanti fra loro ad esempio dati con codice F01 fino a F09, ma non vengono accettati intervalli di tipo F01 fino a F01.....o intervalli di tipo F03 a F01....nel secondo codice sono risolti e sono solo dovuti ad una mia svista sui dati del database...

Cambierò ancora il programma per aggiungere altre scelte di ricerca...

Per prima cosa aumento le possibilità di inserire intervalli nel form di richiesta in questo modo (IMMAGINE 6-6B) :



IMMAGINE 6

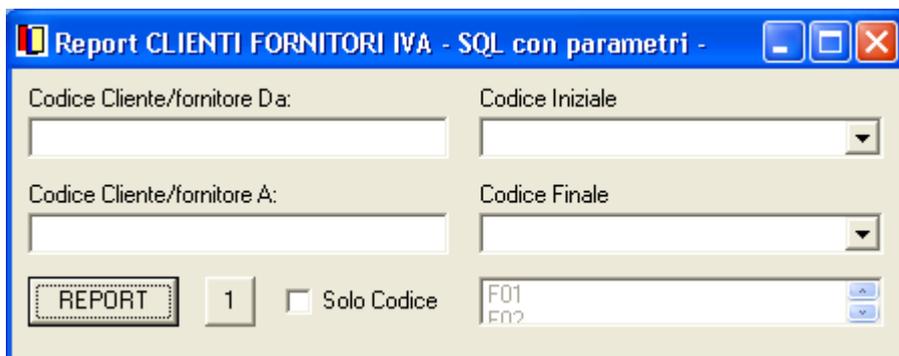


IMMAGINE 6B

Inserisco una lista e un checkbox per vagliare nella ricerca un solo valore sveltendo la ricerca fra codice iniziale e codice finale. Poi inserisco un pulsante che mi permetterà di passare da una SQL con parametri a una SQL senza parametri per verificare il corretto funzionamento dell'una e dell'altra, sul risultato o meno del report. Il pulsante indicato con la captin 1 equivale ad una ricerca con parametri, indicato con il numero 2 ad una ricerca senza parametri ( FIGURA 7 ).



IMMAGINE 7

Facendo una seconda routine che inserisca direttamente tutto il codice SQL, i risultati non cambiano, comunque ecco il codice....

```
function TfmAllegati.Apriqry2: Integer;
var// sostitutivo di aprigry include in se' tutta la SQL
// per cui bisogna cancellarla da QryAllegati
stringa:string;

begin //1
  dmallegati.qryallegati.sql.clear;

  if (rb.Checked = true) then //A
    begin //2
      stringa := 'select C.cod_cli, C.rag_sociale,R.tot_documento, R.imponibile, R.iva from
CLIFOR C left outer join REGIVA R on (R.COD_CLI = C.COD_CLI) where (C.COD_CLI
=' + lbc.Items[lbc.ItemIndex] + ') order by C.COD_CLI, R.ANNO, R.TIPO';
      dmallegati.QryAllegati.sql.clear;
      dmAllegati.QryAllegati.sql.add(stringa);
      showmessage(dmAllegati.QryAllegati.sql.text);
      dmAllegati.QryAllegati.Open;
      result:=dmAllegati.QryAllegati.RecordCount;
      exit;
    end //2
  else //A
    begin //3
      if (cbda.ItemIndex > -1) and (cba.ItemIndex > -1) then //B
        begin //4
          stringa := 'select C.cod_cli, C.rag_sociale,R.tot_documento, R.imponibile, R.iva
from CLIFOR C left outer join REGIVA R on (R.COD_CLI = C.COD_CLI) where
(C.COD_CLI >=' + cbda.Items[cbda.ItemIndex] + ') and (C.COD_CLI <=' +
cba.Items[cba.ItemIndex] + ') order by C.COD_CLI, R.ANNO, R.TIPO';
          dmallegati.QryAllegati.sql.clear;
          dmAllegati.QryAllegati.sql.add(stringa);
          showmessage(dmAllegati.QryAllegati.sql.text);
          dmAllegati.QryAllegati.Open;
          result:=dmAllegati.QryAllegati.RecordCount;
          exit;
        end //4
      else //B // solo testo
        begin //5
          if (cfda.Text <> '') and (cfa.Text <> '') then //C
```

```

        begin //6
            stringa := 'select C.cod_cli, C.rag_sociale,R.tot_documento, R.imponibile, R.iva
from CLIFOR C left outer join REGIVA R on (R.COD_CLI = C.COD_CLI) where
(C.COD_CLI >="" + cfda.text + "") and (C.COD_CLI <="" + cfa.text + "") order by
C.COD_CLI, R.ANNO, R.TIPO';
            dmAllegati.QryAllegati.sql.clear;
            dmAllegati.QryAllegati.sql.add(stringa);
            showMessage(dmAllegati.QryAllegati.sql.text);
            dmAllegati.QryAllegati.Open;
            result:=dmAllegati.QryAllegati.RecordCount;
            exit;
        end //6
    else //C
        begin //7
            result:= 0;
            exit;
        end; //7

    end //5
end //3

end; //1

```

Ed ecco il codice del nuovo pulsante...e quello del report

```

procedure TfmAllegati.ButtonAllegatiClick(Sender: TObject);
begin
if (ButtonAllegati.caption = '1' ) then
begin
tipoqry := 2;
ButtonAllegati.caption := '2';
fmallegati.Caption := 'Report CLIENTI FORNITORI IVA - SQL senza parametri -'
end
else
begin
tipoqry := 1;
ButtonAllegati.caption := '1';
fmallegati.Caption := 'Report CLIENTI FORNITORI IVA - SQL con parametri -'
end;
end;

procedure TfmAllegati.Button1Click(Sender: TObject);
begin
if (tipoqry = 1) then // con sql statica e parametri dinamici

if (apriqry >0) then
begin
ppReport1.device:=dvScreen;
ppReport1.print;
end
else

```

```

begin
  showmessage('Anteprima non disponibile');
end
else //2 con sql dinamica
if (apriqy2 >0) then
  begin
  ppReport1.device:=dvScreen;
  ppReport1.print;
  end
else
  showmessage('Anteprima non disponibile');
end;

```

Una ricerca all' interno del database come in FIGURA 7B ad esempio mi dara' come risultato la figura 7C...



IMMAGINE 7B

Codice	Ragione Sociale	
F0100	...	0
F0101	...	0
F0102	...	549.71
F0103	...	1375.88
F0104	...	0
F0105	...	0

IMMAGINE 7C

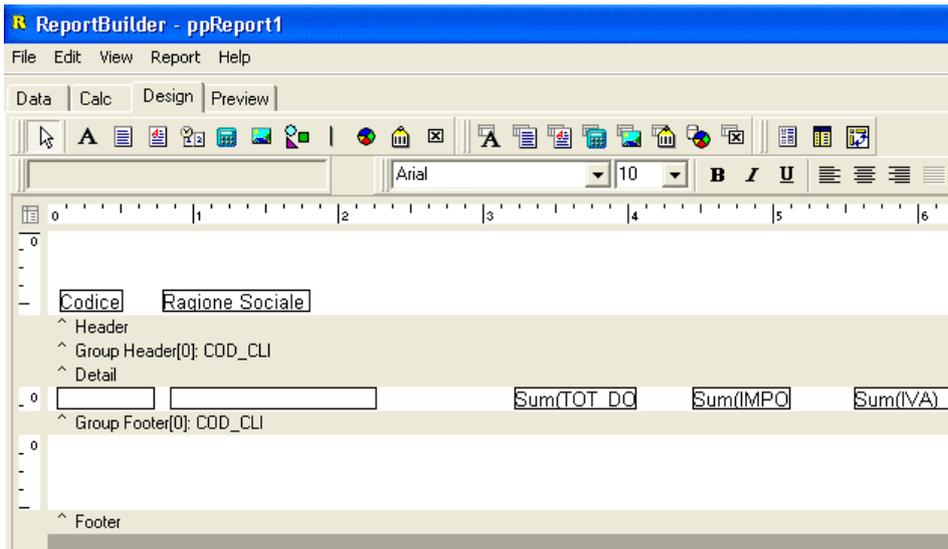


IMMAGINE 8

Torniamo ora al nostro PPREPORT che si presenta come in figura 8...

Da ben individuare sono i campi Sum(Totale documento) Sum(Imponibile) e Sum(Iva). Nella finestra CALC ( IMMAGINE 9 ) ecco cosa troviamo, la descrizione dei tipi di dato presenti nella finestra DESIGN.

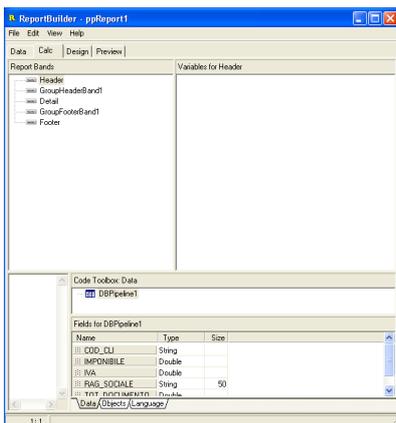


IMMAGINE 9

Per far si che non appaiano tutte le voci riguardante un solo codice ( Cliente o fornitore che sia ) ma solo la somma aritmetica di tutti gli inserimenti,

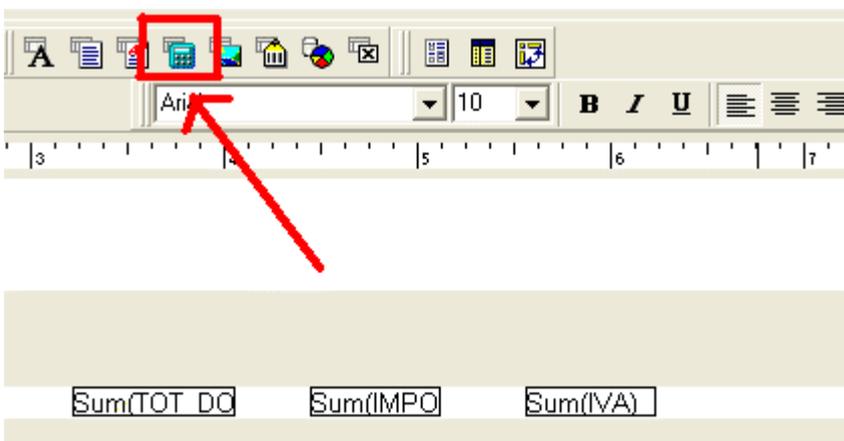
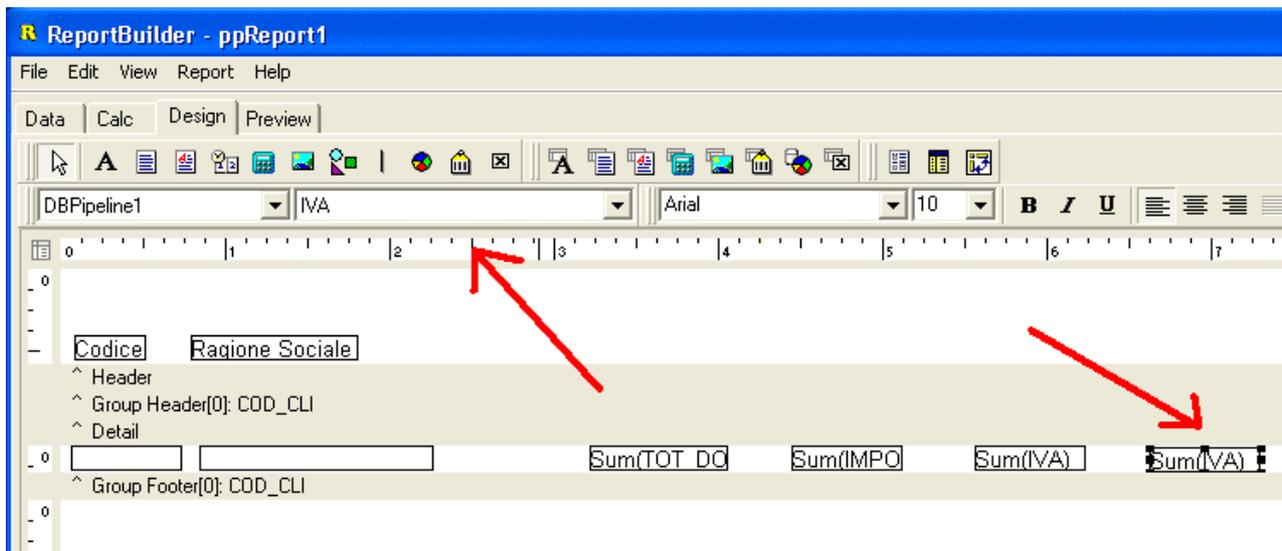


IMMAGINE10

bisogna utilizzare il componente DBcalc che è inserito nel PPreport...

Questo componente permette di fare la somma di tutti i valori di un unico codice in modo da accorciare la lista e di non renderla ridondante verso un singolo soggetto... (IMMAGINE 10).

Per prima cosa si deve inserire un nuovo DBcalc che all' inizio risulterà vuoto con un punto interrogativo a segnalare questo stato. Per inserire il valore che a noi serve ed ottenerne quindi la somma si deve passare il campo attraverso il combobox a discesa vicino a DBpipeline (IMMAGINE 11), in questo caso ho aggiunto un altro campo iva per fare un esempio...



Ecco in breve come si compila un PPreport partendo da una SQL.....Una volta completato il programma ricordarsi di togliere i vari showmessage che ho inserito per la verifica immediata dell' SQL.

### Problematiche con JOIN

Join si può trovare in diversi modi:

semplicemente come **inner join .... On ....** Funziona collegando le due tabelle ed estrae solo i records che fra le 2 tabelle si ritrovano ad avere dei valori uguali, che noi abbiamo posto come verifica es...

```
SELECT
    campi
FROM
    prima_tabella
INNER JOIN
    seconda_tabella
ON
    prima_tabella.chiave_primaria = seconda_tabella.chiave_esterna
```

Con **LEFT OUTER JOIN** invece vengono restituiti tutti i record della tabella di sinistra anche se non corrispondono **ON** alla sequenza di confronto.

```
SELECT
    campi
FROM
    prima_tabella
LEFT OUTER JOIN
    seconda_tabella
    ON
        prima_tabella.chiave_primaria = seconda_tabella.chiave_esterna
```

Per ultimo il caso di uso di **RIGHT OUTER JOIN** vengono restituiti tutti i campi della seconda tabella anche se non corrispondono alla sequenza di controllo ( **ON** ).