conga Support℠

Advanced Conga Queries

After creating a query with the Conga® Query Builder, you may edit the SELECT statement (in the Salesforce® record) to add SOQL features that aren't currently available from the Query Builder.

A popular reason to do so is to include one or more of the new SOQL aggregation functions such as SUM( ), COUNT( ), COUNT_DISTINCT(), AVG( ) and GROUP BY which allow you to group and summarize a collection of records.

Once a query has been manually edited, do not use the Query Builder again on that record because the manual edits are overridden. Use Select Count(ID) from the Opportunity

Due to the complicated nature of SOQL statements, our support staff can only assist with the creation of SOQL statements generated through our tools, such as Conga Query Builder.

Please consult the Salesforce SOQL documentation for complete details about the aggregation functions.

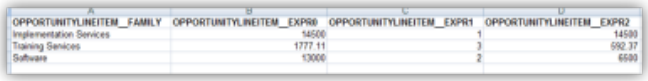
Product Family Example

The following example returns, for each unique Product Family value:

- The sum of Unit Prices
- A count of records
- The average of Unit Price values

```
SELECT PricebookEntry.Product2.Family,
   SUM(UnitPrice),
   COUNT(UnitPrice),
   AVG(UnitPrice)
FROM OpportunityLineItem
WHERE Opportunity.Id = '{pv0}'
GROUP BY PricebookEntry.Product2.Family
```

For a given Opportunity, this query results in the following values in the Conga ViewDataWorkbook.xls file:



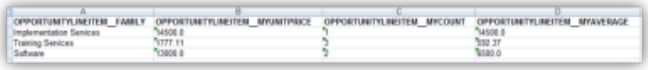
	A	B	C	D
1	OPPORTUNITYLINEITEM__FAMILY	OPPORTUNITYLINEITEM__EXPR0	OPPORTUNITYLINEITEM__EXPR1	OPPORTUNITYLINEITEM__EXPR2
2	Implementation Services	14500	1	14500
3	Training Services	1777.11	3	592.37
4	Software	13000	2	6500

Use Aliases with Aggregate Functions

As displayed above in the Conga ViewDataWorkbook.xls file, the default field names do not clearly identify the source of the values displayed in columns B, C and D. To address this, add to the query a preferred "alias" that will replace the string "EXPRx" and render easily-identifiable field names.

```
SELECT PricebookEntry.Product2.Family
   SUM(UnitPrice) MyUnitPrice
   COUNT(UnitPrice) MyCount
   AVG(UnitPrice) MyAverag
FROM OpportunityLineItem
WHERE Opportunity.Id = '{pv0}'
GROUP BY PricebookEntry.Product2.Family
```

For a given Opportunity, this query results in the following values in the Conga ViewDataWorkbook.xls file:



More Powerful Select Statements: Semi-Join and Anti-Join Support

Semi-Joins use the IN (...) operator within a WHERE clause to use the result of one SELECT statement as the search criteria for another SELECT.

Anti-Joins use the NOT IN (…) operator.

Example 1:

```
SELECT Id, Name
FROM Account
WHERE Id IN
(SELECT AccountId FROM Opportunity WHERE StageName = 'Closed Lost')
```

Example 2:

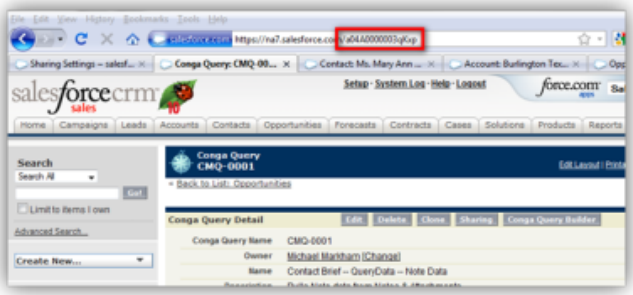
```
SELECT Id
FROM Account
WHERE Id NOT IN (SELECT AccountId FROM Opportunity WHERE IsClosed = false)
```

Include a Conga Query in a Conga Composer button or link

The final step of the process is to include the Id of the Conga Query record in your Conga Composer® button or link.

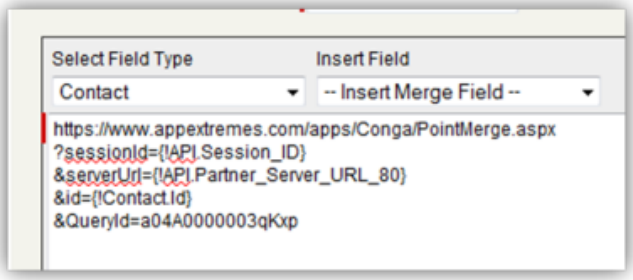
Here's how:

1. Copy the Salesforce Record Id of the Conga Query record.



2. Edit your Conga Composer button.

   Append the &QueryId= parameter, and paste the record ID.



3. (Optional) Include a “Query Alias”:

   You can supply an alias for a Report (or Query) to override the default name.

   Follow these guidelines when adding an alias:

   - The alias must precede the Salesforce Query ID.
   - It must be enclosed in square brackets ("[" and "]")
   - It may contain only alphanumeric characters; punctuation marks and other
   - special characters are not permitted
   - It must contain at least 2 characters and but not more than 20 characters
   - It must be unique within the entire Conga Composer URL
   - Spaces are not permitted within an alias, or between the brackets and the Report Id
   - Reserved sheet names (“Master”, “Org”, “User”, “ReportData”, etc.) may not be used as an alias

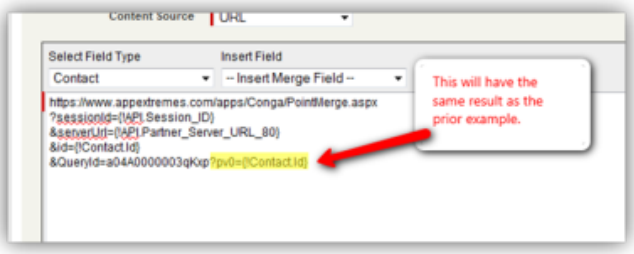
   Example:

   > &QueryId=[MyContacts]a04A000000003qKxp

   The resulting sheet in the View Data Workbook appear labeled as “MyContacts” and would be referenced with this alias in TableStart and TableEnd fields.

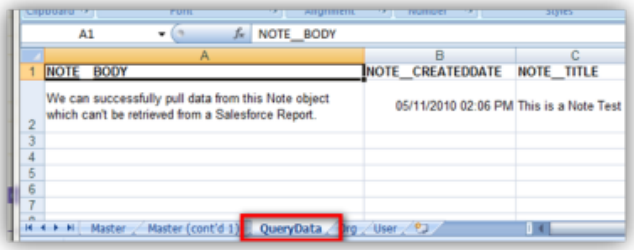
4. (Optional) Specify the values to pass into the Conga Query.

In the absence of a specific value, Conga will automatically pass the master object id into the pv0 field.



5. Save the button.

   Navigate to a Salesforce record that displays the button and click the button. Click the View Data link to see the results.



6. Add the newly available merge fields to your template.