Argos DataBlock Designer Training

Banner Report Specifications

Advanced Exercises

Checkboxes

Notes

Checkboxes are for Boolean inputs, yes or no. Checkboxes are used independently or in conjunction with other controls. Unlike the button control, which has fixed values for the clicked and unclicked state (1 and null); you can define the values of the checked and unchecked state for the checkbox.

Exercise 5 Student Course List (modification)

Michelle reviews the DataBlock and wants you to give her the option of not seeing the dropped courses in the report. Sometimes she wants to see dropped classes but most of the time she doesn't need to see those classes.

Report Description

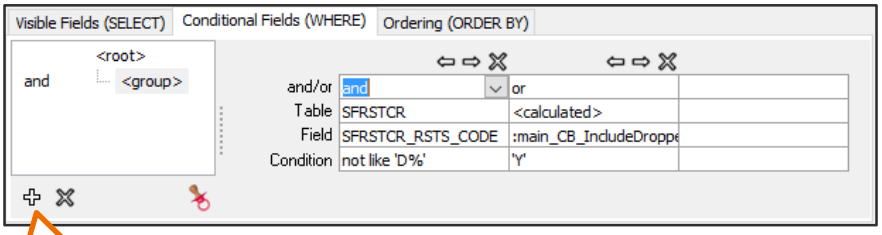
Add a variable that allows the Report Viewer the ability to choose if they see dropped classes in the course listing.

Form Objects

Include Dropped Classes	
Control Type	CheckBox
Variable Name	Main_CB_IncludeDropped
Checked Value	Y
Unchecked Value	N

Report Query

Where		
Conditional Group		
SFRSTCR	SFRSTCR_RSTS_CODE	Not Like 'D%'
OR		
calculated	Main_CB_IncludeDropped	='Y'



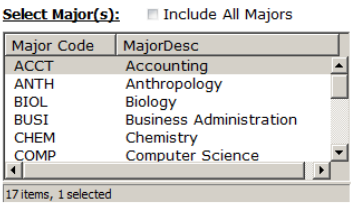
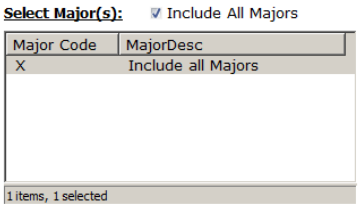
When using checkboxes always think "conditional group"

Notes

Exercise 6 Review "All with Checkbox" example

The "All with Checkbox" technique allows the user to choose between selecting from the list box control to constrain the output dataset or not selecting from the list box control and including all the values in the output dataset, essentially not constraining the output dataset by the list box control.

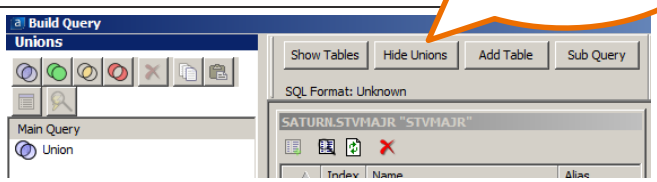
How the user uses the checkbox and listbox

Unchecked: User selects from list	**Checked**: User does not make a selection, everything is automatically selected
	

How to control the listbox with the checkbox

Create a union.

Clicking the Show/Hide Unions button reveals or hides the Unions pane.

How the union looks in the visual designer.	

The SQL.

```
select STVMAJR.STVMAJR_CODE "MajorCode",
       STVMAJR.STVMAJR_DESC "MajorDesc"
  from SATURN.STVMAJR STVMAJR
 where STVMAJR.STVMAJR_VALID_MAJOR_IND = 'Y'
       and :Main_CB_IncludeAllMajors = 'N'
union
select DUAL.DUMMY,
       'Include all Majors'
  from SYS.DUAL DUAL
 where :Main_CB_IncludeAllMajors = 'Y'

 order by 2
```

Describe what the SQL is doing. _____

_____

_____

Notes

How many fields are selected in the main query? _____

How many fields are selected in the **union** query? _____

Why? _____

_____

The order by is set to the number 2 instead of MajorCode. Why? _____

_____

Steps to implement the "All with Checkbox" technique

1. Create checkbox control
    a. Set values for checked and unchecked
2. Link checkbox to list box control
    a. Create union
    b. In main query set condition to return values when checkbox is unchecked
    c. In union query set condition to return string indicating all values are selected when checkbox is checked
    d. Set Order By to column number
3. Set property of list box control to Auto Select
4. Modify the query to use the checkbox and list box in combination
    a. If necessary, remove the original constraint condition that used the list box as a constraint from the root query
    b. Add in a condition group
    c. Add in a constraint to equal the list box value
    d. Add in a constraint to equal the checked value for the checkbox
    e. Set the checkbox constraint to "or"

Notes

Exercise 7 Use "All with Checkbox" in Address List for Admitted Students

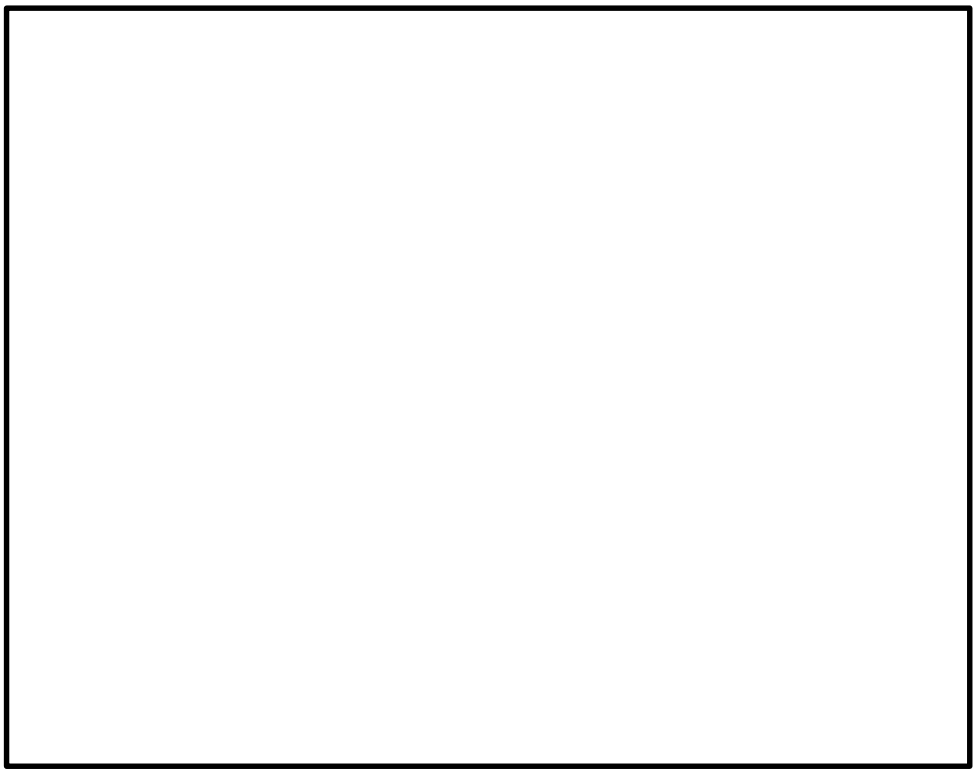
Bethany comes back and wants you to make it easier to select all the admission acceptance types. She says that most of the time when she runs this report she wants all the types and she would like it to default to a select all.

Report Description

Modify the DataBlock form to make the variable selection easier for the Report Viewer by adding a checkbox to switch the admission acceptance list box between showing all the available selections and an "all" choice.

Change - Add a checkbox to select all admission acceptance types.

Dashboard Prototype



Notes

Form Objects

Include All Types		
Control Type	Checkbox	
Variable Name	main_CB_AdmitType	
Checked Value	Y	
Unchecked Value	N	

Admission Acceptance Type		
Control Type	List Box	
Variable Name	main_LB_AdmitAcceptType	
Root Query		
Table	STVADMT	
Select	STVADMT_CODE	as Code
	STVADMT_DESC	as Desc
Where	:main_CB_AdmitType = 'N'	
Order By	2	
Union Query	Union	
Table	Dual	
Select	DUMMY	
	'Include All Acceptance Types'	
Where	:main_CB_AdmitType = 'Y'	

Sort by the column position, instead of the field.

Report Query

Where		
Conditional Group		
SARADAP	SARADAP_ADMT_CODE	=main_LB_AdmitAcceptType.AdmissTypeCode
OR		
calculated	:main_CB_AdmitType	='Y'

Notes

Notes

Exercise 8 Visible/Invisible Object Property

Objects have a visible property. Objects can be always visible (Yes), always invisible (No) or have their visibility dependent upon the value of a SQL variable. Variable functions follow:

- Boolean
  - True: object is visible
  - False: object is invisible
- Integer
  - Non-zero: object is visible
  - Zero: object is invisible
- String
  - Non-blank: object is visible
  - Blank: object is invisible
- SQL
  - Non-blank: object is visible
  - Blank: object is invisible

Report Description

To demonstrate the visible property modify the Budget Summary DataBlock to show the dashboard button only when all the required variables have been selected. This can be done easily by showing the button only when the last variable, account is not null.

Form Objects

Run Dashboard	
Control Type	Button
Variable Name	Main_BT_RunDashbd
Visible	Main_LB_Acct.AcctCode

Exercise 9 Enable/Disabled Object Property

The enable/disabled property specifies whether the object is always enabled (Yes), always disabled (No), or the enable/disable state is determined by a value of a variable. Variable functionality is the same as with the visible/invisible property.

Report Description

As a demonstration exercise, modify the Student List DataBlock "Get Students" button to only be enabled if there is an entry in either the ID edit box or the Name edit box. This will alleviate the potential problem of a user searching for all students in a term, which takes too much time.

SQL Variable

SQL_ButtonEnabled		
Control Type	SQL variable	
Variable Name	SQL_ButtonEnabled	
Table	DUAL	
Fields	'True'	as TrueValue
Where	:main_DD_Term.STVTERM_CODE	is not null
Conditional Group		
	:main_EB_IDNumber	is not null
OR		
	:main_EB_LastName	is not null

Form Objects

Get Student List	
Control Type	Button
Variable Name	Main_BT_RunDashbd
Enabled	SQL_ButtonEnabled.TrueValue

Notes

Free Type

Exercise 10 Convert visual design to Free Type

The Free Type button 

in the toolbar allows you to convert your visual designer to free type SQL. **Once a SQL statement is converted to Free Type it can not be converted back to the visual design.**

Exercise

Copy the Address List DataBlock, along with its report, and convert the Report Query to Free Type.

Exercise 11 Add Special Characters to report query

Once the report query has been converted to Free Type the reports can not be modified to add in additional filters or change the sort order unless special comments are added into the report query SQL.

Exercise

Add in special characters to enable Filters and Sorts to be applied at the report level.

Best Practice: Add Special Characters to the report query before publishing the DataBlock.

Notes

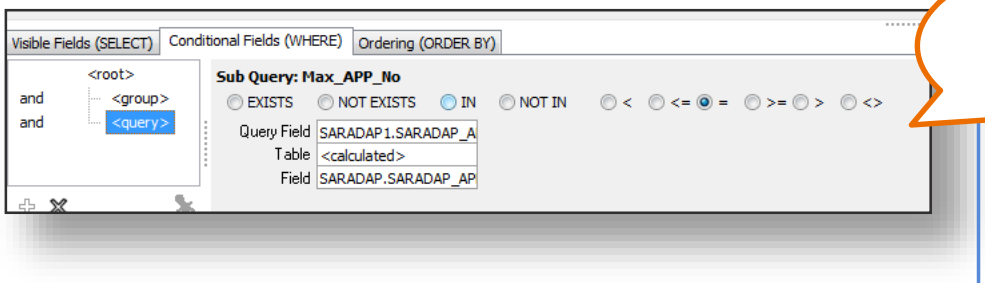
Subqueries

Exercise 12 Correlated Subquery in the Where clause

The data in our Address List report query dataset is returning data for every application. Students sometimes submit multiple applications so we need to restrict our data so that it only contains the most current information. Add a subquery to return one row per student by constraining the result set to max application number.

Report Query

Where		
Subquery SARADAP_APPL_NO =		
Select	Max (SARADAP_APPL_NO)	
From	SARADAP	
Where	SARADAP1_PIDM	= SARADAP_PIDM
	SARADAP1_TERM_CODE_ENTRY	= SARADAP_TERM_CODE_ENTRY



When you add the query, keep the table row set to calculated. Set the field row to the appropriate field.

Exercise 13 Scalar Subquery in the SELECT clause

Remember to remove the validation tables if you have them already added to the query.

Report Query

Select	
(select stvlevl_desc from stvlevl where saradap_levl_code = stvlevl_code and rownum = 1)	LevelDesc
(select stvadmt_desc from stvadmt where SARADAP.SARADAP_ADMT_CODE = stvadmt_code and rownum = 1)	AdmitDesc

This is Part 1 of a two-part exercise. Part 1 covers the DataBlock Designer responsibilities. Part 2 is covered in the Report Writer training class and covers the responsibilities of the Report Writer.

Notes

To link datasets you must have a common unique identifier in both the detail dataset and the sub-detail dataset.

Datasets

Adding datasets at the report level allows you to connect to multiple databases or to do multiple groups, instead of nested grouping. Datasets also allow for the addition of charts to a banded report.

To link datasets you must have a common unique identifier (including datatype) in both the detail dataset and the sub-detail dataset.

Exercise 14 Add Datasets to Address List for Admitted Students banded report (Part 1)

Bethany needs to be able to view email addresses and phone numbers for each admitted student in her banded report. If she includes the email address and phone number in the report query each unique combination would create a detail row.

Name	Address	Phone	Email
John Smith	Address 1	Phone 1	Email 1
John Smith	Address 1	Phone 1	Email 2
John Smith	Address 1	Phone 2	Email 1
John Smith	Address 1	Phone 2	Email 2

If she wants to group the phone numbers and address together, she will need to create two datasets, one to get the phone numbers and another to get the email addresses.

John Smith	Address 1
	Phone 1
	Phone 2
	Email 1
	Email 2

Report Description

Modify the banded report to add two datasets linked to the Report Query so the Report Writer can display the email addresses and phone numbers for each student.

Material

Your trainer will give you the DataBlock and report to use for this exercise.

- DataBlock: Advanced – Datasets
- Report: Subdetail_Designers Start Here

Notes

Report Dataset

Email Addresses		
Control Type	Dataset	
Variable Name	SQL_GetEmail	
Table	GOREMAL	
Select	GOREMAL_EMAL_CODE	As EmailCode
	GOREMAL_EMAIL_ADDRESS	As EmailAddress
	GOREMAL_STATUS_IND	As StatusInd
	GOREMAL_PREFERRED_IND	PreferredInd
Where	GOREMAL_PIDM = :ArgosData.InternalId	
Order By	GOREMAL_EMAL_CODE Ascending	

Phone Numbers		
Control Type	Dataset	
Variable Name	SQL_Phone	
Table	SPRTELE	
Select	SPRTELE_TELE_CODE	TeleCode
	SPRTELE_PHONE_AREA	AreaCode
	SPRTELE_PHONE_NUMBER	PhoneNo
	SPRTELE_STATUS_IND	StatInd
	SPRTELE_ATYP_CODE	OptionAddrType
Where	SPRTELE_PIDM= :ArgosData.InternalId	
Order By	SPRTELE_TELE_CODE Ascending	

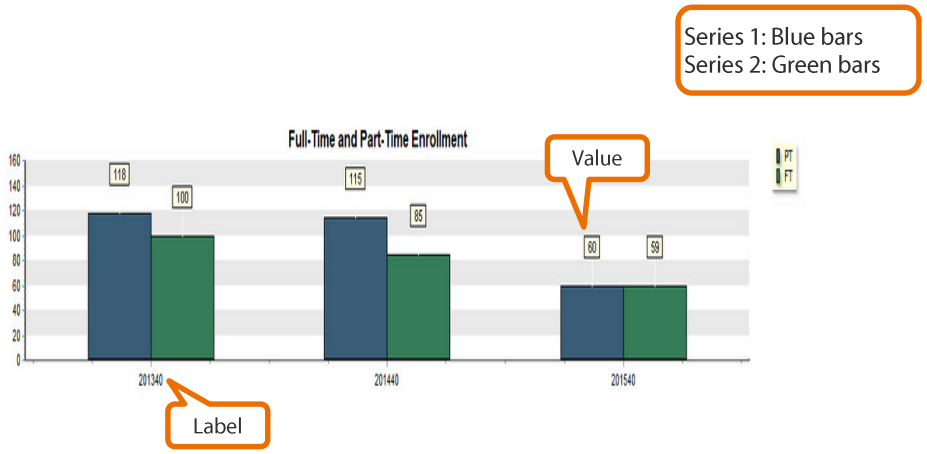
The common, unique identifier used to link the datasets is the PIDM in this query.

Notes

Exercise 15 Add dataset to banded report for chart

Definitions

- Series: A series is any group of related data items that you want to plot on a chart. A series is based upon a dataset.
- Value: The numerical field that is being charted.
- Label: The field that describes what is being charted.



Report Description

Bethany wants a count of all the different students by admit type and she wants it grouped together instead of a count at the bottom of each group. You suggest to her that a chart would be an easy way to display the information to the readers in a summary form.

Material

Your trainer will give you the DataBlock and report to use for this exercise.

- DataBlock: Advanced – Datasets
- Report: Chart_Designers Start Here

Report Dataset

The easiest and quickest way for you to create the dataset will be to copy the report query and modify it to count PIDMs and group by admit description.

Admit Types	
Control Type	Dataset
Variable Name	SQL_AdmitType
Query	Copy the report query & modify
Select	COUNT SPRIDEN_ID
	GROUP BY STVADMT_DESC

Notes

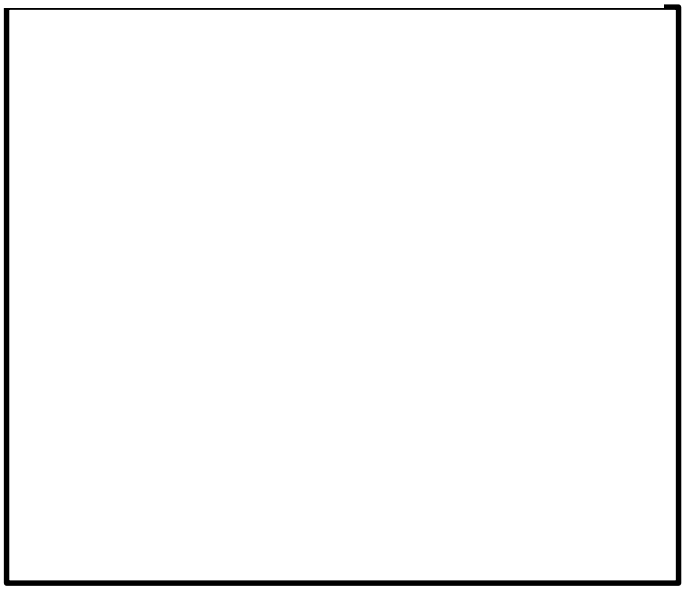
Exercise 16 Registered Students Analytical Dashboard

Report Description

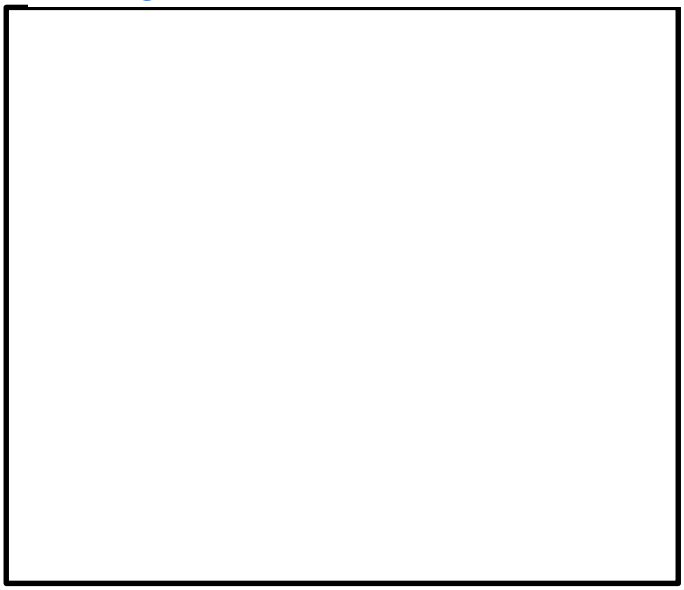
This DataBlock allows the user to view analysis of registered student data by academic term. There are four forms to display with charts displaying overall student count and broken down by age range, gender, and ethnicity; and an OLAP cube.

Dashboard Prototype

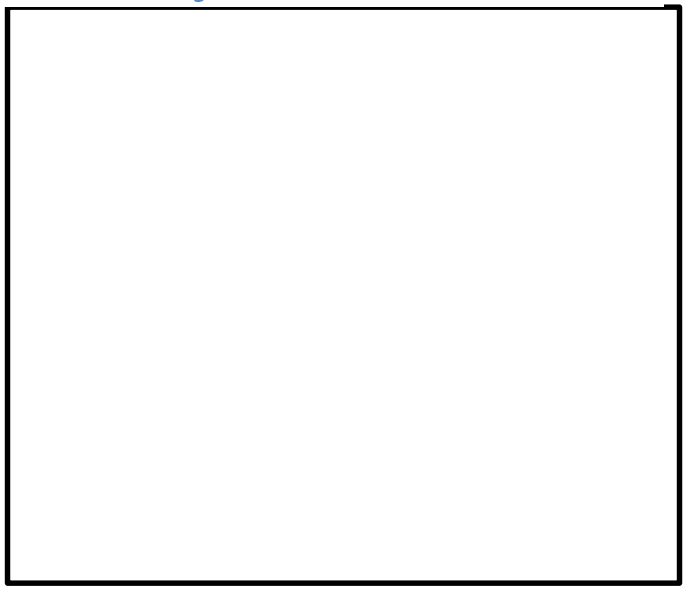
Form 1: Variable Selection



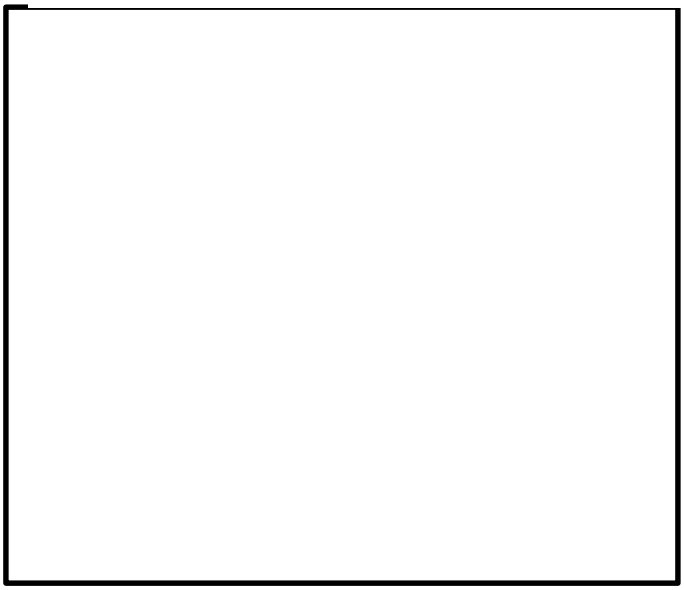
Form 2: Registered Students Chart



Form 3: Drill Through Charts



Form 4: OLAP



Form Objects: Main

Academic Year		
Control Type	Drop Down Box	
Variable Name	main_DD_AcadYr	
Table	STVTERM	
Select **Distinct**	STVTERM_ACYR_CODE	as AcadYr
Where	STVTERM_CODE	<> '000000'
	STVTERM_START_DATE	<= sysdate + 60
Order By	STVTERM_ACYR_CODE	Descending

Term Type			
Control Type	Radio Button		
Variable Name	main_RB_TermType		
Table	STVTERM		
Select **Summing**	Group By	Substr(STVTERM_CODE,5,2)	as TermType
Where	STVTERM_CODE		<> '000000'
Order By	1		Descending

Notes

Form Objects: Chart

Run Registered Student Chart	
Control Type	Button
Variable Name	Chart_BT_RunChart

Registered Students				
Control Type	Chart			
Variable Name	Chart_CT_RegStdnts			
Series Name	RegStudents			
Dataset Name	SQL_RegStudents			
Table	SFBETRM			
Select **Summing**	Count	SFBETRM_PIDM	as InternalID	
	Group By	SFBETRM_TERM_CODE	as Term	
Where	Substr(SFBETRM_TERM_CODE,1,4)	between :main_DD_AcadYr.AcadYr – 5 And :main_DD_AcadYr.AcadYr		
	Substr(SFBETRM_TERM_CODE,5,2)	=:main_RB_TermType.TermType		
	F_registered_this_term(SFBETRM_PIDM,SFBETRM_TERM_CODE)	= 'Y'		
	:chart_BT_RunChart	is not null		
Order By	SFBETRM_TERM_CODE	Descending		

Notes

Form Objects: ChartB

Gender Chart

Gender				
Control Type	Chart			
Variable Name	ChartB_CT_Gender			
Series Name	Gender			
Dataset Name	SQL_Gender			
Tables	SFBETRM, SPBPERS			
Select **Summing**	Count	SFBETRM_PIDM	as InternalID	
	Group By	case when SPBPERS.SPBPERS_SEX = 'F' then 'Female' when SPBPERS.SPBPERS_SEX = 'M' then 'Male' else 'Unknown' end	as Gender	
Where	Inner join	SFBETRM_PIDM	= SPBPERS.SPBPERS_PIDM	
	f_registered_this_term(SFBETRM.SFBETRM_PIDM,SFBETRM.SFBETRM_TERM_CODE)	='Y'		
	SFBETRM.SFBETRM_TERM_CODE	= :chart_CT_RegStdnts.RegStudents.Term		
Order By	<nothing>			

Notes

Ethnicity Chart

Ethnicity				
Control Type	Chart			
Variable Name	ChartB_CT_Ethnicity			
Series Name	Ethnicity			
Dataset Name	SQL_Ethnicity			
Tables	SFBETRM, SPBPERS, STVETHN			
Select **Summing**	Count	SFBETRM_PIDM	as InternalID	
	Group By	STVETHN_DESC	as EthnicGroup	
Where	Inner join	SFBETRM_PIDM	= SPBPERS_PIDM	
	Inner join	SPBPERS_ETHN_CODE	= STVETHN_CODE	
	f_registered_this_term(SFBETRM.SFBETRM_PIDM,SFBETRM.SFBETRM_TERM_CODE)		='Y'	
	SFBETRM.SFBETRM_TERM_CODE		= :chart_CT_RegStdnts.RegStudents.Term	
Order By	STVETHN_DESC		Ascending	

Notes

Age Range Chart (Independent Exercise)

Age Range				
Control Type	Chart			
Variable Name	ChartB_CT_Age			
Series Name	Age			
Dataset Name	SQL_Age			
Tables	SFBETRM, SPBPERS			
Select **Summing**	Count	SFBETRM_PIDM		as InternalID
Group By	case when SPBPERS.SPBPERS_BIRTH_DATE is null then 'Unknown' when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY'))< 18 then 'Under 18' when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY')) < 23 then '18 to 22' when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY')) < 26 then '23 to 25' else 'Over 25' end			as Age_Group
Group By	case when SPBPERS.SPBPERS_BIRTH_DATE is null then 1 when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY'))< 18 then 2 when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY')) < 23 then 3 when (substr (SFBETRM.SFBETRM_TERM_CODE, 1, 4)) - (to_char (SPBPERS.SPBPERS_BIRTH_DATE, 'YYYY')) < 26 then 4 else 5 end			As Sort_Me
Where	join	SFBETRM_PIDM	= SPBPERS.SPBPERS_PIDM	
		f_registered_this_term(SFBETRM.SFBETRM_PIDM,SFBETRM.SFBETRM_TERM_CODE)	='Y'	
	SFBETRM.SFBETRM_TERM_CODE	= :chart_CT_RegStdnts.RegStudents.Term		
Order By	3	Ascending		

Notes

Form Objects: OLAP

Button

Run OLAP Cube	
Control Type	Button
Variable Name	OLAP_BT_RunOLAP

OLAP Cube

OLAP			
Control Type		OLAP	
Variable Name		Keep default	
Tables	SFBETRM, SGBSTDN		
Select	SFBETRM_PIDM		as InternalID
	SFBETRM_TERM_CODE		Term
	SFBETRM_ESTS_CODE		EnrollStat
	SFBETRM_RGRE_CODE		RegReasonCode
	SGBSTDN_LEVL_CODE		EffectLevel
	SGBSTDN_STYP_CODE		StudTypeCode
	SGBSTDN_MAJR_CODE_1		Curr1MajorCode
	(select stvdegc_desc from stvdegc where sgbstdn_degc_code_1 = stvdegc_code and rownum = 1)		Degree
	f_get_desc_fnc('STVCOLL',SGBSTDN.SGBSTDN_COLL_CODE_1,30)		College
Where	Outer Left Join	SFBETRM_PIDM	= SGBSTDN_PIDM
	f_registered_this_term(SFBETRM.SFBETRM_PIDM,SFBETRM.SFBETRM_TERM_CODE)		='Y'
	Chart_BT_RunOLAP		is not null
	substr(SFBETRM.SFBETRM_TERM_CODE,1,4)		between :main_DD_AcadYr.AcadYr -5 and :main_DD_AcadYr.AcadYr
	substr(SFBETRM.SFBETRM_TERM_CODE,5,2)		=:main_RB_TermType.TermType
Subquery SGBSTDN_TERM_CODE_EFF=			
Table	SGBSTDN		as SGBSTDN1
Select	MAX	SGBSTDN_TERM_CODE_EFF	as EffectTermStudRec
Where	SGBSTDN1.SGBSTDN_PIDM		=SGBSTDN.SGBSTDN_PIDM
Order By	<nothing>		

Notes

Do not set Order Bys or Group Bys in an OLAP cube. Let the OLAP cube do the sorting and grouping.

Independent Exercise – Add Checkbox to Budget Summary

Adrian likes that the Budget Summary DataBlock now allows him to look at one or two accounts at a time on the dashboard. He does not like having to select all the account numbers from the list box when he wants to run a report since that is what he usually does.

Report Description

Modify the DataBlock form to add in a checkbox to select all account codes to make running reports easier for Adrian

Steps

1. Add checkbox.
    a. What value will you set for checked? _____
    b. What value will you set for unchecked? _____
2. Modify list box.
    a. What new SQL clause needs to be added? _____
    b. Explain what the new conditions in the SQL clause will be. _____

    _____

    _____

    _____

    _____

    c. How will the data be sorted? _____
3. Modify the multicolumn list box.
    a. What condition needs to be removed from the root WHERE clause?

    _____

    b. What type of clause needs to be added to the WHERE clause?

    _____

    c. Describe the new conditions that are added to the WHERE clause.

    _____

    _____

    _____

4. What changes need to be made to the report query?

    _____

    _____

Notes

Form Objects

Include All Accounts	
Control Type	Checkbox
Variable Name	main_CB_Acct
Checked Value	Y
Unchecked Value	N

Account			
Control Type	List Box		
Variable Name	main_LB_Acct		
Root Query			
Table	FGBBAVL		
Select **Summing**	Group By	FGBBAVL_ACCT_CODE	As AcctCode
Where	:main_CB_Acct = 'N'		
Order By	1		
Union Query	Union		
Table	Dual		
Select	'Include All Account Codes'		
Where	:main_CB_Acct = 'Y'		

Report Query

Where		
Conditional Group		
FGBBAVL	FGBBAVL_ACCT_CODE	=main_LB_Acct.FGBBAVL_ACCT_CODE
OR		
calculated	:main_CB_Acct	='Y'

Notes