

Advanced Query Techniques

VICKY WALL

AND

LENNA WENKE

Welcome!

At the end of this course users will:

- Be able to add criteria to a left outer join.
- Utilize subqueries to effectively search for query data information.
- Efficiently find record and field information using the front end of PeopleSoft in conjunction with other tools.
- Gain the knowledge and skills to check your query results to be confident of query accuracy.
- Effectively apply aggregate functions to query results.
- Utilize the Having tab to add criteria to fields using aggregate functions.
- Be able to create expressions to allow complex refining of query results.
- Understand BIND records and how they affect query development.

Standard Joins

Standard joins only display rows where there is a match between BOTH records, meaning it is possible not all of the rows from Record A will be displayed.

Record A	
Name	ID
Mike	001
Sophia	002
Olivia	003
Drake	004
Amanda	005
Caroline	006
Josh	007
Lucinda	008
Carol	009
Alan	010
Brett	011



Record B	
ID	Favorite Thing
001	Fishing Pole
002	Computer
003	Playing
004	Dirt
006	Dogs
007	Engines
008	Antiques
009	Reading
011	Football



Standard Join		
ID	Name	Favorite Thing
001	Mike	Fishing Pole
002	Sophia	Computer
003	Olivia	Playing
004	Drake	Dirt
006	Caroline	Dogs
007	Josh	Engines
008	Lucinda	Antiques
009	Carol	Reading
011	Brett	Football

Outer Join

This join type will always display all of the rows in Record A. It will simply display BLANK where there is no match.

Record A	
Name	ID
Mike	001
Sophia	002
Olivia	003
Drake	004
Amanda	005
Caroline	006
Josh	007
Lucinda	008
Carol	009
Alan	010
Brett	011



Record B	
ID	Favorite Thing
001	Fishing Pole
002	Computer
003	Playing
004	Dirt
006	Dogs
007	Engines
008	Antiques
009	Reading
011	Football



Outer Join		
ID	Name	Favorite Thing
001	Mike	Fishing Pole
002	Sophia	Computer
003	Olivia	Playing
004	Drake	Dirt
005	Amanda	
006	Caroline	Dogs
007	Josh	Engines
008	Lucinda	Antiques
009	Carol	Reading
010	Alan	
011	Brett	Football

Example: Normal Outer Join

Suppose we wanted to have a list of all our enrolled GRAD students for Fall 2016 and we want to know if they have a mobile phone number stored in our system.

Start with your base query: Add the STDNT_CAR_TERM record and define your enrollment criteria:

Records Query Expressions Prompts Fields **Criteria** Having Dependency Transformations View SQL Run

Query Name: New Unsavd Query Description: Feed

Add Criteria Group Criteria Reorder Criteria

Logical	Expression1	Condition Type	Expression 2	Edit	Delete
	A.STRM - Term	equal to	2167	Edit	
AND	A.UNT_TAKEN_PRGRSS - Units Taken for Progress	greater than	0	Edit	
AND	A.ACAD_CAREER - Academic Career	equal to	GRAD	Edit	

Save Save As New Query Preferences Properties Publish as Feed Publish as Pivot Grid New Union

Next, look up the PERSONAL_PHONE record. We are going to outer join this record to STDNT_CAR_TERM. When you click the “Join Record” link, select the outer join option on the next screen:

Select join type and then record to join with PERSONAL_PHONE - Personal Data - Phone Numbers.

Join Type

Join to filter and get additional fields (Standard Join)

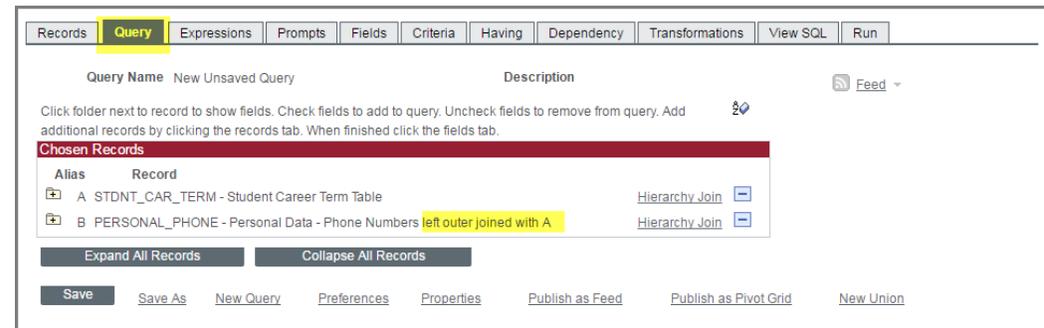
Join to get additional fields only (Left outer join)

Join Record Personalize | Find | First 1 of 1 Last

A = STDNT_CAR_TERM - Student Career Term Table

Cancel

You should now notice a couple of differences in your query. The first is on your Query tab: the phone record has additional information next to its name, indicating that it is outer joined and which record it is joined to:



The second is on your Criteria tab. Any criteria for outer joins has an additional column filled in: “Belongs to”.

Records Query Expressions Prompts Fields **Criteria** Having Dependency Transformations View SQL Run

Query Name New Unsaved Query Description Feed

Add Criteria Group Criteria Reorder Criteria

Logical	Expression 1	Condition Type	Expression 2	Edit	Delete	Belongs to
	A.STRM - Term	equal to	2167	Edit	[-]	
AND	A.UNT_TAKEN_PRGRSS - Units Taken for Progress	greater than	0	Edit	[-]	
AND	A.ACAD_CAREER - Academic Career	equal to	GRAD	Edit	[-]	
AND	A.EMPLID - Empl ID	equal to	B.EMPLID - Empl ID	Edit	[-]	B

Save Save As New Query Preferences Properties Publish as Feed Publish as Pivot Grid New Union

This new criteria column will become important when we narrow our results. Go back to your Query tab and add new criteria on the PHONE_TYPE field in the phone record. Since this is an outer join, you will have to change the “belongs to” field at the bottom to the same letter as the record (in our case, since we are adding criteria to record B, the criteria should belong to outer join B). Click “OK”.

Edit Criteria Properties

Choose Expression 1 Type

Field
 Expression

Expression 1

Choose Record and Field

Record Alias.Fieldname

- Phone Type

*Condition Type

Choose Expression 2 Type

Field
 Expression
 Constant
 Prompt
 Subquery

Expression 2

Define Constant

Constant

This criteria belongs to

OK Cancel

Correct outer join

099842849	MOBL
099858619	MOBL
099952539	MOBL
099957689	MOBL
011033399	
098924269	
011425916	
011258929	
011160689	
011225946	

Criteria have been erroneously placed on the WHERE clause.

000689470	MOBL
000714020	MOBL
000726080	MOBL
000779490	MOBL
009936448	MOBL
009985400	MOBL
010012858	MOBL
010013065	MOBL
010033968	MOBL

Results incorrectly display only MOBL

Add Criteria to Your Left Outer Join

Adding Criteria to a Left Outer Join – Simplified Instructions

- Create a query that has a left outer join, select a field from the record which has been left outer joined and add criteria to it, either from the Criteria tab or the Add Criteria icon.
- In the “Criteria Belongs To” box select the outer join clause that matches the record alias (for example, Record B).
- Select fields from both records.

Criteria page when you are using a subquery

Records Query Expressions Prompts Fields **Criteria** Having Dependency Transformations View SQL Run

Query Name New Unsaved Query Description [Feed](#)

Working on selection Top Level of Query [Subquery/Union Navigation](#)

[Add Criteria](#) [Group Criteria](#) [Reorder Criteria](#)

Logical	Expression 1	Condition Type	Expression 2	Edit	Delete
	ASTRM - Term	equal to	2197	Edit	-
AND	AACAD_CAREER - Academic Career	equal to	GRAD	Edit	-
AND		exists	SUBQUERY	Edit	-

[Save](#) [Save As](#) [New Query](#) [Preferences](#) [Properties](#) [Publish as Feed](#) [Publish as Pivot Grid](#) [New Union](#)

Query page when you are creating a subquery

Records **Query** Expressions Prompts Fields Criteria Having Dependency Transformations View SQL Run

Query Name New Unsaved Query Description [Feed](#)

Working on selection **Subquery for exists** [Subquery/Union Navigation](#)

Add additional records by clicking the records tab. When finished select a single field for this subquery and you will be transferred to the fields tab.

Alias	Record
B STDNT_ENRL - Student Enrollment Table	Hierarchy Join

Fields	Find	View All	First	1-50 of 69	Last
Select EMPLID - Empl ID					Join PEOPLE_SRCH - People Search View
Select ACAD_CAREER - Academic Career					Join STDNT_CAREER - Student Career
Select INSTITUTION - Academic Institution					Join INSTITUTION_TPI -

SubQueries

Edit Criteria Properties

Choose Expression 1 Type

Field
 Expression

Expression 1

Choose Record and Field

Record Alias.Fieldname
A.EMPLID - ID

*Condition Type in list

Choose Expression 2 Type

In List
 Subquery

Expression 2

Define Subquery

Define/Edit Subquery

OK Cancel

2 common types of criteria conditions for sub-queries.

1. Using in list
2. Using exists which does not require using a field.

The 2 look very different.

Edit Criteria Properties

*Condition Type exists

Choose Expression 2 Type

Subquery

Expression 2

Define Subquery

Define/Edit Subquery

OK Cancel

Expression Basics

Step 1: Create literal value as a column

The image shows a screenshot of the Microsoft Access Expressions tab in a query design view. The 'Query Name' is 'New Unsaved Query'. The 'Add Expression' button is highlighted with a red box and a callout box that says 'Click Add Expression'. Below the button, it says 'No expressions have been defined yet.' To the right, the 'Edit Expression Properties' dialog box is open. It has several fields and buttons:

- '*Expression Type:' dropdown menu set to 'Character' (highlighted with a red box and callout '3. Select Expression Type').
- 'Length:' text box set to '20' (highlighted with a red box and callout '4. Set length of expression').
- 'Aggregate Function' checkbox (unchecked).
- 'Decimals:' text box (empty).
- 'Expression Text:' text area containing the text 'has a dollar value of' (highlighted with a red box and callout '5. Enter Expression Text in single quotation marks').
- 'Add Prompt' and 'Add Field' buttons (unselected).
- 'OK' and 'Cancel' buttons at the bottom (the 'OK' button is highlighted with a red box and callout '6. Click OK').

Expression Operator Basics

Delimiter	Meaning	Delimiter	Meaning
+	Addition operator	<>	Relational operator (not equal)
'	Character string delimiter	<	Relational operator (less than)
	Concatenation operator	>	Relational operator (greater than)
/	Division operator	<=	Relational operator (less than or equal)
(Expression or list beginning delimiter	>=	Relational operator (greater than or equal)
)	Expression or list end delimiter	;	Statement terminator
,	Item Separator	-	Subtraction operator
=	Relational operator (equal)	*	Multiplication operator

Expressions

Expressions are calculations that PeopleSoft Query performs as part of a query. Use them when you must calculate a value that PeopleSoft Query does not provide by default—for example, to add the values from two fields together or to multiply a field value by a constant.

Use Expressions in two ways:

- As comparison values in selection criteria.
- As columns (fields) in the query output.

Edit Expression Properties

***Expression Type**
Signed Number ▼ Length

Aggregate Function Decimals

Expression Text

```
CASE WHEN SUM(B.XREF_AMT) IS NULL THEN  
A.ITEM_AMT ELSE (A.ITEM_AMT - SUM(B.XREF_AMT))  
END
```

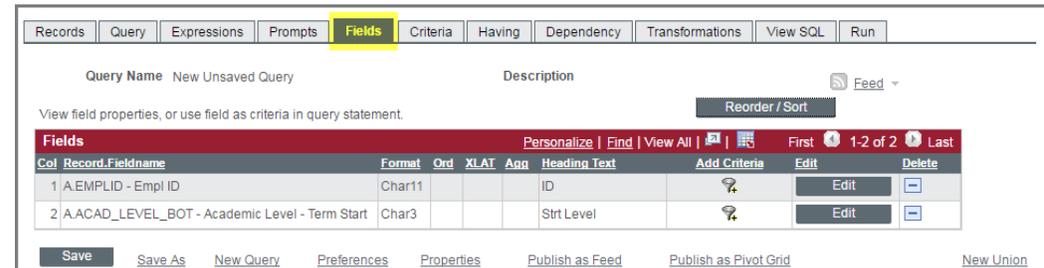
[Add Prompt](#) [Add Field](#)

Expression Basics

Example: Translating Values

Sometimes, all you want to know is “Does this person have something in this field or not?” or “What does this funny combination of numbers and letters mean in plain English?” You don’t care about the details that the system stores; you just want something that you can glance at and easily make sense of. The following will allow you to create an expression that will “translate” the system values into something that you, the query writer, define.

Suppose we wanted to look at STDNT_CAR_TERM and count the number of students enrolled in each class standing (freshman, sophomore, junior, etc.) for Fall 2016. Create your new query with STDNT_CAR_TERM and add your fields:



Click the “Edit” button next to the EMPLID field and apply the count aggregate function

Edit Field Properties

Field Name A.EMPLID - Empl ID

Heading	Aggregate
<input type="radio"/> No Heading	<input type="radio"/> None
<input type="radio"/> Text	<input type="radio"/> Sum
<input checked="" type="radio"/> RFT Short	<input checked="" type="radio"/> Count
<input type="radio"/> RFT Long	<input type="radio"/> Min
Heading Text ID	<input type="radio"/> Max
*Unique Field Name A.EMPLID	<input type="radio"/> Average

OK Cancel

Next, add your criteria:

Records Query Expressions Prompts Fields **Criteria** Having Dependency Transformations View SQL Run

Query Name New Unsaved Query Description Feed

Add Criteria Group Criteria Reorder Criteria

Logical	Expression1	Condition Type	Expression 2	Edit	Delete
	ASTRM - Term	equal to	2167	Edit	
AND	AUNT_TAKEN_PGRSS - Units Taken for Progress	greater than	0	Edit	

Save Save As New Query Preferences Properties Publish as Feed Publish as Pivot Grid New Union

Case statement

If you run your query now, the academic level will read “10”, “20”, “30”, and other values that may not make much sense unless you know what each value translates to. Our job is to translate these values for the user, so that they don’t have to do it themselves. Doing this can make the query easier to read at a glance.

For most queries, you will want to remove the field that you are translating from the query. There’s no need to have both the old, hard to read stuff *and* your new stuff. For the sake of the example, however, keep it in if you want to see the “before and after” contrast.

Head over to the Expressions tab and add a new expression.

The easiest way to translate our system values is through a CASE statement expression. They are written like this:

CASE

WHEN A.FIELDNAME = ‘system value 1’ THEN ‘your value’

WHEN A.FIELDNAME = ‘system value 2’ THEN ‘your value’

ELSE ‘default value’

END

Expression Basics

Numeric Step 2: Use field as a column in the result set

Records Query **Expressions** Prompts Fields Criteria Having Dependency Transformations View SQL Run

Query Name: New Unsaved Query Description: Feed

Add Expression

Expressions List Personalize Find 1 of 1 Last

Expression Text	Use as Field	Add Criteria	Edit	Delete
CASE WHEN A.ACAD_LEVEL_BOT = '10' THEN '1' WHEN A.ACAD_LEVEL_BOT = '20' THEN '2' WHEN A.ACAD_LEVEL_BOT = '30' THEN '3' WHEN A.ACAD_LEVEL_BOT = '40' THEN '4' WHEN A.ACAD_LEVEL_BOT = '50' THEN '5' WHEN A.ACAD_LEVEL_BOT = 'GR' THEN '6' WHEN A.ACAD_LEVEL_BOT = 'P1' THEN '7' WHEN A.ACAD_LEVEL_BOT = 'P2' THEN '7' WHEN A.ACAD_LEVEL_BOT = 'P3' THEN '7' WHEN A.ACAD_LEVEL_BOT = 'P4' THEN '7' WHEN A.ACAD_LEVEL_BOT = '80' THEN '8' WHEN A.ACAD_LEVEL_BOT = '' THEN '' ELSE '8' END	Use as Field		Edit	

Save Save As New Query Preferences Properties Publish as Feed Publish as Pivot Grid New Union

View All | Rerun Query | Download to Excel | Download to XML

First 1-12 of 12 Last

	Count ID	Strt Level	CASE WHEN A.ACAD_LEVEL_BOT =
1	551	08	8
2	5650	10	1
3	4653	20	2
4	6325	30	3
5	7202	40	4
6	417	50	5
7	3585	GR	6
8	630	P1	7
9	654	P2	7
10	182	P3	7
11	277	P4	7
12	130	IAL	8

Using Expressions vs Criteria

Choose Expression 1 Type

Field
 Expression

Expression 1

Define Expression

Expression A.NET_AWARD_AMT/4

[New Expression](#) [Edit the Expression](#)

*Condition Type not less than

Choose Expression 2 Type

Field
 Expression
 Constant
 Prompt
 Subquery

Expression 2

Define Constant

Constant 500

Choose Expression 1 Type

Field
 Expression

Expression 1

Define Expression

Expression 'X'

[New Expression](#) [Edit the Expression](#)

*Condition Type equal to

Choose Expression 2 Type

Field
 Expression
 Constant
 Prompt
 Subquery

Expression 2

Define Expression

Expression 'X' AND 0 <> INSTR(';' || :2 || ';' || RTRIM(A.EMPLID) || ',')

[Add Prompt](#) [Add Field](#)

Advanced Functions

Function	Usage	Example	Result
NVL	Allows you to substitute a value when a NULL value is encountered.	Example 1: NVL(SUPPLIER_CITY, 'Please Complete')	If Supplier City is populated this will return the city name, if NULL it will return 'Please Complete'.
NVL2	Extends the functionality found in the NVL function. Substitute a value when NULL is encountered as well as when a non-NULL value is encountered.	Example 1: NVL2(SUPPLIER_CITY, 'Completed', 'Please Complete')	If Supplier City is populated this will return 'Completed', if NULL it will return 'Please Complete'.
ROWNUM	Assigns a number indicating the order in which each row is returned by a query.	ROWNUM	2931
ROWNUM	Limits the number of rows returned in a result set.	ROWNUM < 10	Returns 9 rows of results.
CASE	Performs the functionality of an "IF-THEN-ELSE" statement CASE WHEN condition_1 THEN result_1 WHEN condition_2 THEN result_2 ELSE result END	CASE WHEN A.ACAD_CAREER = 'UGRD' THEN 'Undergraduate Student' WHEN A.ACAD_CAREER = 'GRAD' THEN 'Graduate Student' ELSE 'Check Career' END	Undergraduate Student

Expression Gotchas

Case statements –

When there is a possibility of not meeting your specified condition(s) which can cause multiple rows to be returned for the same entry. To get around this you can use Aggregates such as minimum or maximum to specify which value you want to return.

ROWNUM – this is not an absolute value so if you specify a ROWNUM not greater than 200 you may for example get 210 rows.

Any time you are working with number there is a potential for data distortion. You can get around this by using a sum or count expression but that may also eliminate rows that you may need.

There are a few cases where you can still get incorrect data using this type of expression. An example is when data distortion occurs in the STDNT_AWRD_DISB record, where a student has multiple disbursements of the same dollar amount. The DISTINCT keyword will keep the first row but discard all others, resulting in a dollar amount that is too small. Be on the lookout if you think that this type of error is a possibility in your query. Removing the DISTINCT from your expression will bring back the data distortion, so there is no way around this issue.

Bind Records

E SCC_PS_STRM_BND - PopSel Bind record for STRM [Hierarchy Join](#) 

Check All Uncheck All

Fields		Find View All	First  1-5 of 5  Last
<input type="checkbox"/>	 EMPLID - Empl ID	Join PEOPLE_SRCH - People	 +
<input type="checkbox"/>	 ACAD_CAREER - Academic Career	Search View Join STDNT_CAREER - Student Career	 +
<input type="checkbox"/>	 STRM - Term	Join TERM_TBL - Term Definition Table	 +
<input type="checkbox"/>	 INSTITUTION - Academic Institution	Join INSTITUTION_TBL - Institution Table	 +
<input type="checkbox"/>	NAME - Name		 +

Pop Select Query Protocol

Pop Select Queries have the following protocol:

1. Identify the Query/Batch Process relationship. If unsure what the relationship is submit a ticket to Enterprise Systems to ensure that the correct Query is tied to the correct process or look at a query that is already using a BIND record to see what it is using.
2. All key fields in the BIND record must be displayed in the Query results.
3. Only display the BIND record key fields in the Query results. a. In some batch processes if any other column is displayed besides a BIND record key field it will cause the process to fail. If it is desired to use the Pop Select Query for data values in addition to its use in the batch process it is recommended to create two separate Queries.
4. Always include POP or POPSEL in the Query name at the end of the Query. For example, WSU_SF_MIS_NATID_POPSEL.
5. Describe the Query as a Pop Select Query in the description.
6. Provide a full Definition, including the batch process the Query is tied to.

Edit Field Properties



Field Name A.POSTED_TOTAL_AMT - Posted Total Amount

Heading

No Heading RFT Short

Text RFT Long

Heading Text

Sum Posted Total Amount

*Unique Field Name

A.POSTED_TOTAL_AMT

OK

Cancel

Aggregate

None

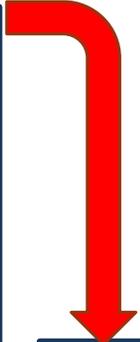
Sum

Count

Min

Max

Average



	Unit	Ledger	Account	Currency	Year	Sum Total Amt	Base Curr
1	WA170	LOCAL	101110	USD	2013	48778576.320	USD
2	WA170	LOCAL	101110	USD	2014	-12406059.840	USD
3	WA170	LOCAL	101110	USD	2015	5830548.110	USD
4	WA170	LOCAL	101110	USD	2016	34659659.960	USD
5	WA170	LOCAL	101120	USD	2013	25080.430	USD
6	WA170	LOCAL	101120	USD	2014	233865.460	USD
7	WA170	LOCAL	101120	USD	2015	-144044.820	USD
8	WA170	LOCAL	101120	USD	2016	14029.750	USD

Aggregates Function

Having

Records Query Expressions Prompts Fields Criteria **Having** View SQL Run

Query Name: New Unsaved Query

Add Having Criteria No having criteria have been added yet.

Click **Add Having Criteria**

Edit Having Criteria Properties

Choose Expression 1 Type

Field
 Expression

Expression 1

Choose Record and Field

Record Alias.Fieldname:
B.MONETARY_AMOUNT - Monetary A

*Condition Type: greater than

Choose Expression 2 Type

Field
 Expression
 Constant
 Prompt
 Subquery

Expression 2

Define Constant

Constant: 10000

OK Cancel

2. Select Expression Type 1

3. Select Condition Type

4. Select Expression Type 2

5. Click OK

Having Criteria Tab

Records Query Expressions Prompts Fields Criteria **Having** View SQL Run

Query Name: New Unsaved Query Description:

Add Having Criteria Group Having Criteria

Having Criteria					
Logical	Expression1	Condition Type	Expression 2	Edit	Delete
<input type="text"/>	B.MONETARY_AMOUNT - Monetary Amount	greater than	10000	Edit	-

Customize | Find | First 1 of 1 Last



Create
Your Own
Query!
