

WASHINGTON STATE  
UNIVERSITY

## 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.
- 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 the rows from Record A will be displayed.

Record A		Record B		Standard	
Name	ID	ID	Favorite Thing	ID	Name
Mike	001	001	Fishing Pole	001	Mike
Sophia	002	002	Computer	002	Sophia
Olivia	003	003	Playing	003	Olivia
Drake	004	004	Dirt	004	Drake
Alexandra	005	006	Dogs	006	Caroline
Caroline	006	007	Engines	007	Joah
Joah	007	008	Antiques	008	Lucinda
Lucinda	008	009	Reading	009	Carol
Carol	009	011	Football	011	Elliott
Alan	010				
Brett	012				

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## Outer Join

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

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

---

---

---

---

---

---

---

---


---

---

WASHINGTON STATE UNIVERSITY

## 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:




---

---

---

---

---

---

---

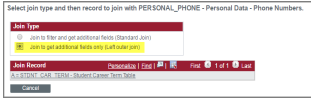
---

---

---

WASHINGTON STATE UNIVERSITY

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:




---

---

---

---

---

---

---

---

---

---

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:

[illegible][illegible]

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

[illegible]

---

---

---

---

---

---

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".

The screenshot shows the 'Edit Criteria Properties' dialog box. The 'Expression 1' tab is selected, displaying a criteria expression: 'Record Aka's Software'. Below the expression is a search icon and a dropdown menu showing 'BPHONE.TYPE', 'Phone Type', and 'Phone Type (tblPhone)'. The 'Expression 2' tab is also visible, showing a criteria expression: 'Contains <NCLL>'. The 'This criteria belongs to' dropdown is set to 'tblCriteria (tblCriteria)'. The 'Criteria' tab is also visible, showing a criteria expression: 'Record Aka's Software'.

[illegible]

WASHINGTON STATE UNIVERSITY

## Correct outer join

849		MOBL
619		MOBL
539		MOBL
589		MOBL
999		
299		
819		
929		
889		
949		

WASHINGTON STATE UNIVERSITY

---

---

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## Criteria have been erroneously placed on the WHERE clause.

470		MOBL
020		MOBL
080		MOBL
490		MOBL
448		MOBL
400		MOBL
858		MOBL
965		MOBL
968		MOBL

WASHINGTON STATE UNIVERSITY

---

---

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## SubQueries

Criteria page when you are using a subquery

Query page when you are creating a subquery

WASHINGTON STATE UNIVERSITY

---

---

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

2 common types of criteria conditions for sub-queries.

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

The 2 look very different.

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## 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.

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## Expression Basics

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## 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

---

---

---

---

---

---

---

---

---

---

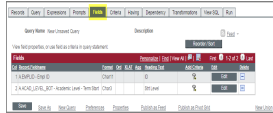
WASHINGTON STATE UNIVERSITY

## Expression Example

**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:




---

---

---

---

---

---

---

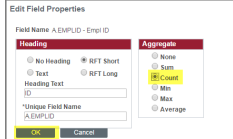
---

---

---

WASHINGTON STATE UNIVERSITY

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




---

---

---

---

---

---

---


---

---

---

WASHINGTON STATE UNIVERSITY

Next, add your criteria:



---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## 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
```

---

---

---

---

---


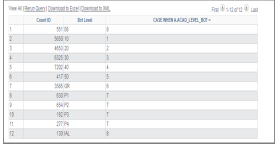
---

---

---

WASHINGTON STATE UNIVERSITY

## Use field as a column in the result set

---

---

---

---

---

---

---

---

7

WASHINGTON STATE UNIVERSITY

## Using Expressions in Criteria

---

---

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## 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 A.ACAD_CAREER = 'UGRAD' THEN 'Undergraduate Student' WHEN A.ACAD_CAREER = 'GRAD' THEN 'Graduate Student' ELSE 'Check Career' END	Undergraduate Student

---

---

---

---

---

---

---

---

---

---

WASHINGTON STATE UNIVERSITY

## Expression Gotchas

**Case statements:**  
When there is a possibility of not meeting your specified condition(s), multiple rows can 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.**

---

---

---

---

---

---

---

---

---

---



E SCC\_PS\_STRM\_BND - PopSelBind record for STRM Hierarchy\_Join

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

[illegible][illegible]

**Edit Field Properties**

Field Name: A:POSTED\_TOTAL\_AMT - Posted Total Amount

**Posting**

- ☒ No Heading
- ☐ # RPT Desc
- ☐ Text
- ☐ # RPT Long
- ☐ Heading Text
- ☐ #
- ☐ Min
- ☐ Max
- ☐ Average

**Aggregation**

- ☐ None
- ☐ Sum
- ☐ Count
- ☐ Std Dev
- ☐ Min
- ☐ Max
- ☒ Average

Unique Field Name: A:POSTED\_TOTAL\_AMT

OK Cancel

	Unit	Ledger	Account	Currency	Year	Sum Total Avc	Base Curr
1	W/A/T	LOCAL	101110	USD	2013	48777676.320 USD	
2	W/A/T	LOCAL	101110	USD	2014	-1249559.840 USD	
3	W/A/T	LOCAL	101110	USD	2015	5303041.110 USD	
4	W/A/T	LOCAL	101110	USD	2016	34685659.890 USD	
5	W/A/T	LOCAL	101120	USD	2013	25500.04 USD	
6	W/A/T	LOCAL	101120	USD	2014	233965.48 USD	
7	W/A/T	LOCAL	101120	USD	2015	-144044.820 USD	
8	W/A/T	LOCAL	101120	USD	2016	-141796.750 USD	

[illegible]

**Having**

1. Click Add Having Criteria

2. Select Expression Type 1

3. Select Condition Type

4. Select Expression Type 2

5. Click OK

---

---

---

---

---

---

---

---

**Having Criteria Tab**

Local	Expression	Condition Type	Expression 2	Edit	Delete
	B MONETARY_AMOUNT - Monetary Amount	greater than	10000	Edit	Delete

---

---

---

---

---

---

---

---

**Create Your Own Query!**

---

---

---

---

---

---

---

---