PeopleSoft Query Reporting Rel 8.50

Volume II - Student Guide

D61552GC10
1.0
December 2009
D64766

ORACLE

Contents

Volume I

Volume II

Lesson 10

Creating Expressions

Objectives

By the end of this lesson, you will be able to:

- Define expressions.
- Create simple expressions.
- Use literals in expressions.
- Use functions in expressions.
- Use expressions in criteria.
- Use links in expressions.
- Use prompts in expressions.
- Use aggregates in expressions.

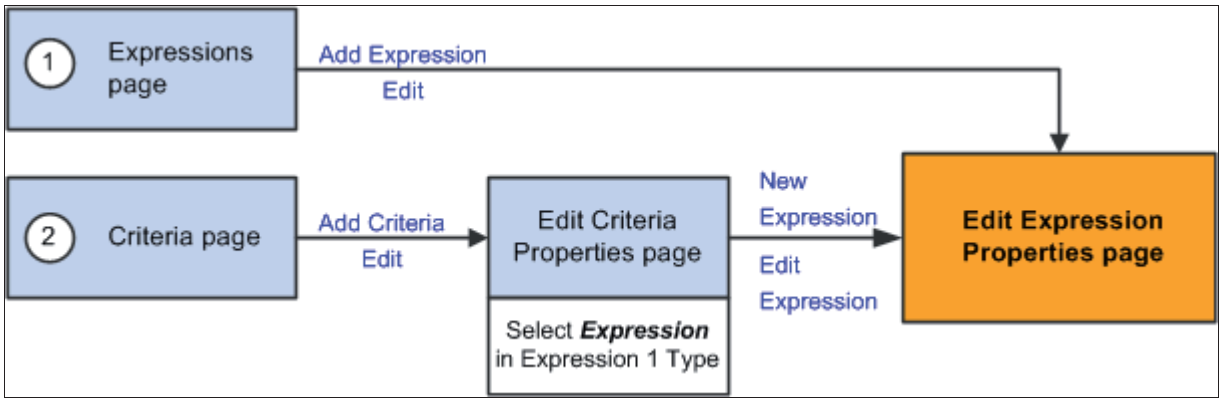
Slide 126

Defining Expressions

Expressions

Expressions are calculations that Query Manager performs as part of a query.

This diagram shows the navigation path to access the Edit Expressions Properties page that you use to define expressions:



Slide 127

Student Notes

Expressions

You use expressions to *display* the value of a field differently than the way you *store* the value; for example, to display the sum of values from two fields, to display the product of a field value and a constant, or to display the values from several fields as one value.

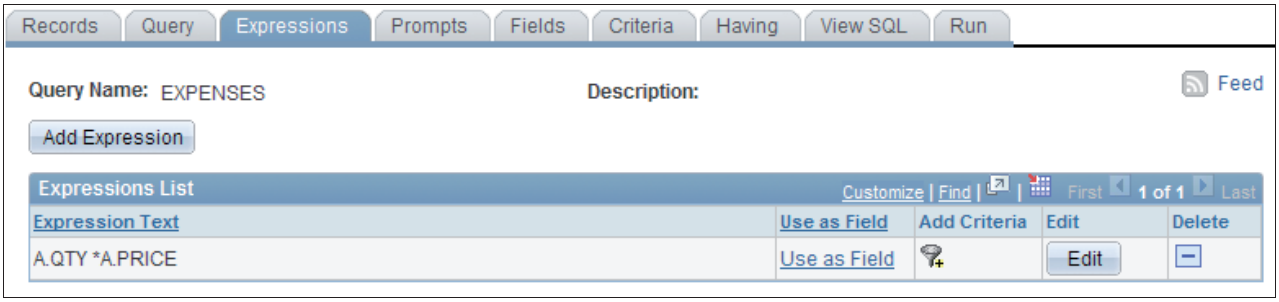
You can work with an expression as if it is a field in the query. You can select it for output, change its column heading, or choose it as an *order-by* column.

Pages Used to Manage Expressions

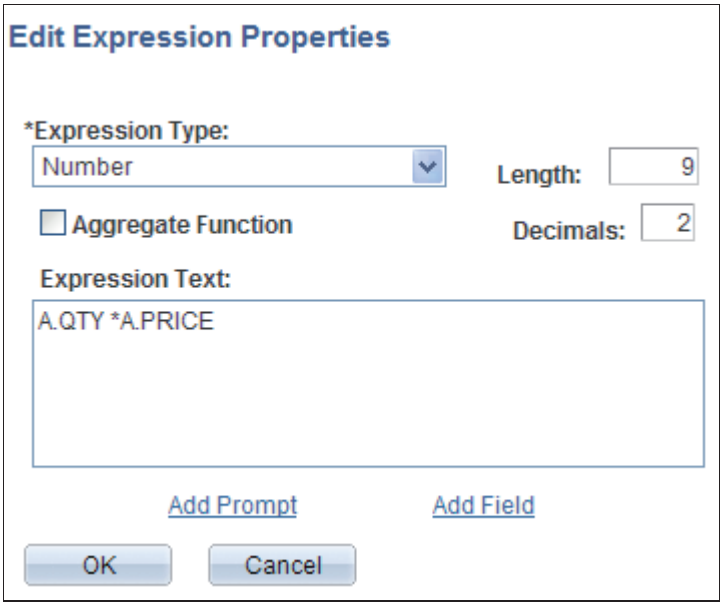
Use these pages to manage expressions in queries:

Page Name	*Navigation*
Expressions	1. Select Reporting Tools, Query, Query Manager. 2. Create a new query or open an existing one. 3. Select the Expressions tab.
Edit Expression Properties	On the Expressions page, click the Add Expression or Edit button.

- Use this page to create and maintain expressions in a central location:



- Use this page to define expression values for queries:



Elements of the Expressions Page

The elements in the Expressions page are:

Add Expression	Click to add new expressions to the query.
Use as Field	Click to use an expression as a field in the query and as a column in the results.
Add Criteria	Click to use an expression as a row of criteria.
Edit	Click to modify existing expressions.
Delete	Click the Delete icon associated with the expression to delete it.

Note. When you delete expressions from the Expressions page, you remove all rows of criteria that reference those expressions.
You can also use the Criteria page to delete rows of criteria that use expressions.

Elements of the Edit Expression Properties Page

The elements in the Edit Expression Properties page are:

Expression Type	Select an appropriate data type. • If you select the *Character* option, enter the maximum length of the expression result in the Length field. • If you select the *Number* or *Signed Number* option, enter the total number of digits in the Length field and the number of digits after the decimal point in the Decimal field.
Length	Enter the length of the expression type.
Aggregate Function	Select to use an aggregate function, such as *Sum, Avg, Count,* and so on. Select if the expression references any field on the Fields page that uses a predefined aggregate function.
Decimals	Enter the number of decimals to appear in the results.
Expression Text	Enter the expression text to perform a calculation; for example, *SUM(A.COST).*
Add Prompt	Click to add existing prompts to this expression.
Add Field	Click to add existing fields to this expression.

Defining Expressions (continued)

Two Ways to Use Expressions

You can use expressions in two ways:

	Customer	Type	Units	Total Cost of Units *125
As columns in the query output.	ABN	FRND	400	50000
	ALBRAS	FRND	1000	125000

	Customer	Type	Units	(Total Cost of Units *125) > 60000
In criteria to filter out data.	ALBRAS	FRND	1000	125000

Slide 128

Student Notes

Two Ways to Use Expressions

You can:

- Use expressions as fields or columns in the query output.

  You can use an expression as a field like any other fields in a query. When you preview the query, the expression name appears as a column heading in the query. You have the option to change its column heading or sort by it.

  In the previous example, the first table shows partial results of a query that uses an expression to calculate total cost of training units: `(TRAINING_UNITS * 125)`.

- Use expressions in criteria to filter out data.

  In the previous example, the second table shows the same query, only this query uses the expression on the Criteria page: `(TRAINING_UNITS * 125, Greater than, 60000)`. The row with 50,000 units is filtered out because the total cost is greater than 60,000.

Defining Expressions (continued)

Adding Expressions as Fields

The first way to use expressions is adding expressions as fields or columns in the query output:

	Customer	Type	Units	Total Cost of Units *125
As columns in the query output.	ABN	FRND	400	50000
	ALBRAS	FRND	1000	125000

	Customer	Type	Units	(Total Cost of Units *125) > 60000
In criteria to filter out data.	ALBRAS	FRND	1000	125000

Slide 129

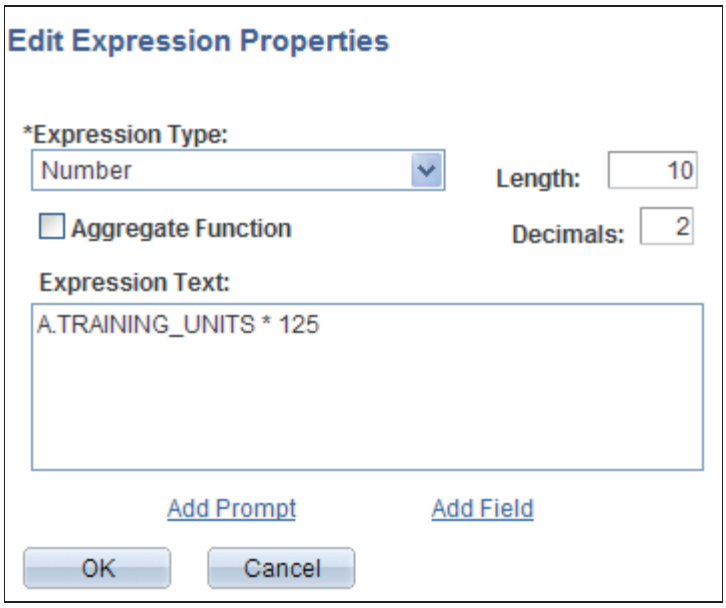
Student Notes

Steps Used to Add Expressions as Fields (Columns)

To add an expression as a field:

1. Use the Edit Expression Properties page to enter or edit the expression.

   In this example, the Training Units expression multiplies the customer's training units by 125 to determine the total cost of training units for each customer:

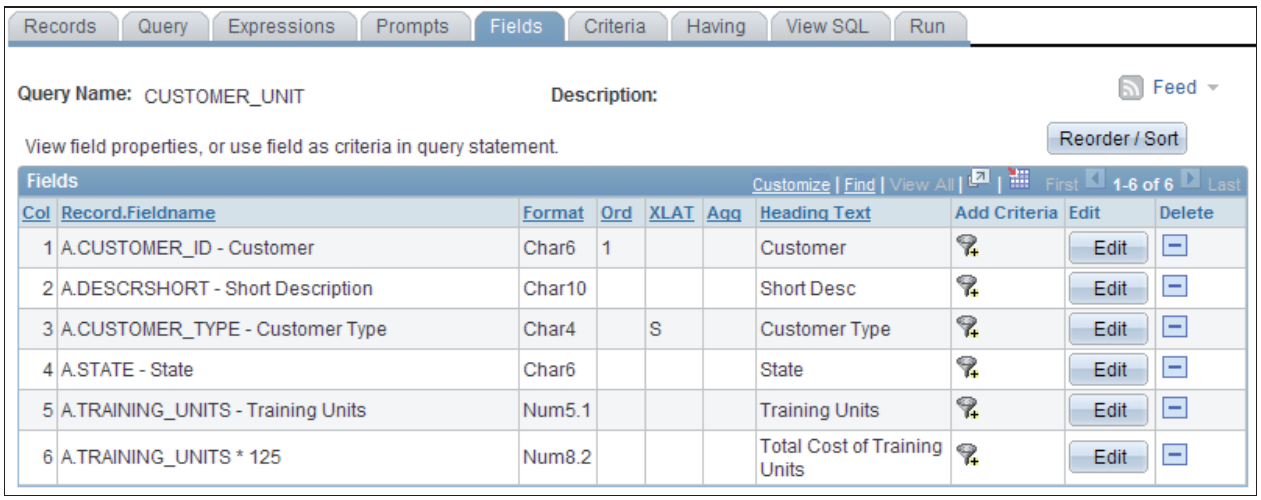


2. In the Expressions page, click the Use as Field link to display the expression on the Fields page and as a column in the result set.

   This example shows the Use as Field link on the Expressions page:

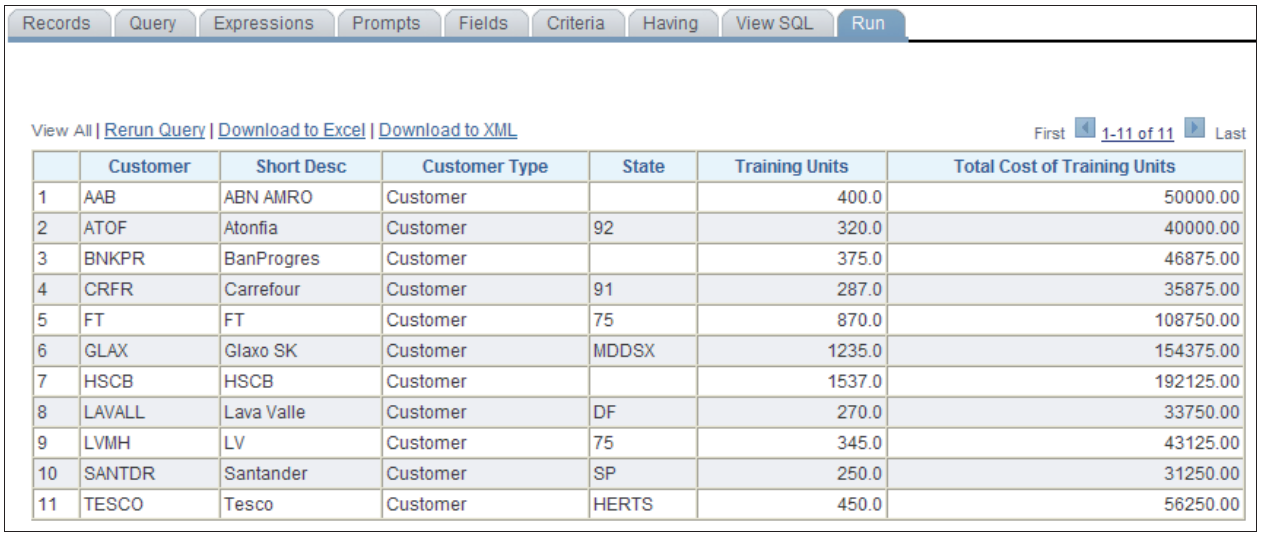
3. The expression appears in the Fields page as a field.

   In this example, column six—TRAINING_UNITS * 125—is an expression used as a field with a descriptive heading text:



4. Save the query, and view the results.

   This example shows the Total Cost of Training Units column, which demonstrates the use of an expression in a column:



	Customer	Short Desc	Customer Type	State	Training Units	Total Cost of Training Units
1	AAB	ABN AMRO	Customer		400.0	50000.00
2	ATOF	Atonfia	Customer	92	320.0	40000.00
3	BNKPR	BanProgres	Customer		375.0	46875.00
4	CRFR	Carrefour	Customer	91	287.0	35875.00
5	FT	FT	Customer	75	870.0	108750.00
6	GLAX	Glaxo SK	Customer	MDDSX	1235.0	154375.00
7	HSCB	HSCB	Customer		1537.0	192125.00
8	LAVALL	Lava Valle	Customer	DF	270.0	33750.00
9	LVMH	LV	Customer	75	345.0	43125.00
10	SANTDR	Santander	Customer	SP	250.0	31250.00
11	TESCO	Tesco	Customer	HERTS	450.0	56250.00

Defining Expressions (continued)

Adding Expressions into Criteria

The second way to use expressions is adding expressions into criteria:

	Customer	Type	Units	Total Cost of Units *125
As columns in the query output.	ABN	FRND	400	50000
	ALBRAS	FRND	1000	125000

	Customer	Type	Units	(Total Cost of Units *125) > 60000
In criteria to filter out data.	ALBRAS	FRND	1000	125000

Slide 130

Student Notes

Steps Used to Add Expressions into Criteria

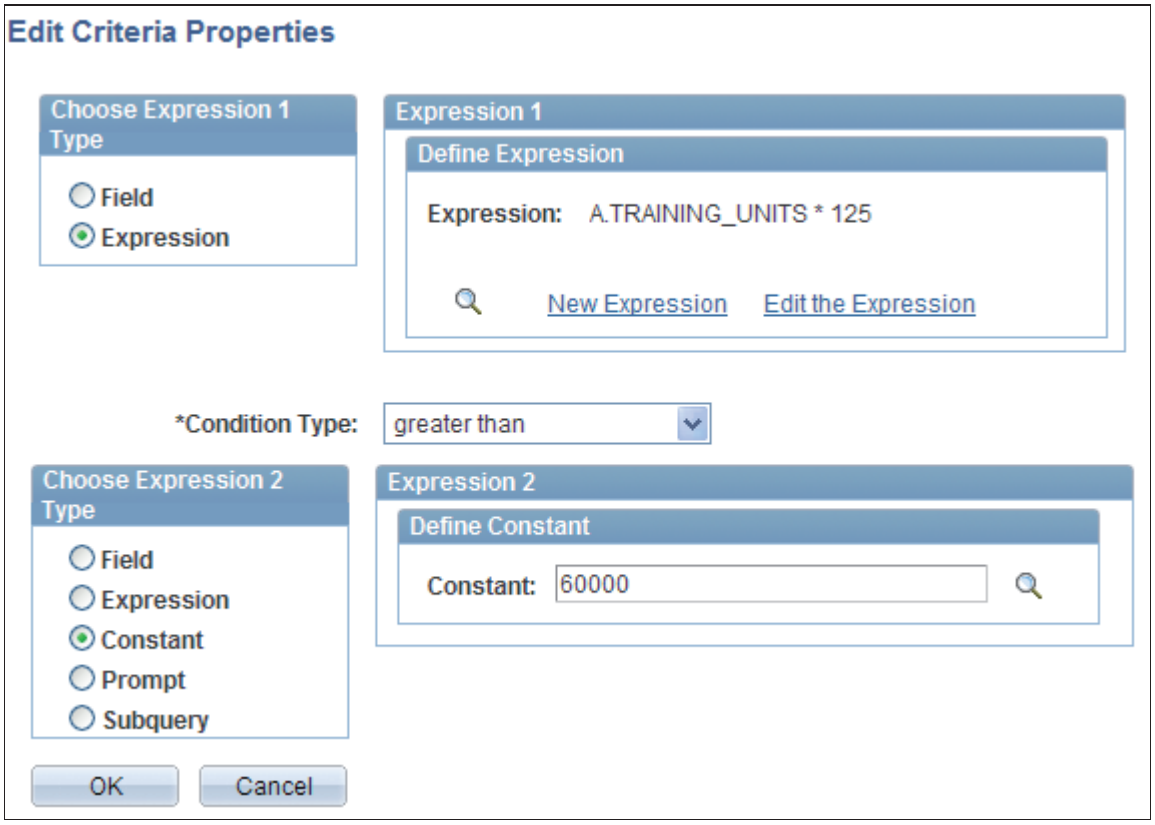
To add an expression into a criteria:

1. Access the Edit Criteria Properties page, and select the Expression option in the Choose Expression 1 Type section.
2. Click the New Expression or Edit the Expression link to access the Edit Expression Properties page.

3. Enter or edit the values of the expression, and click the OK button.

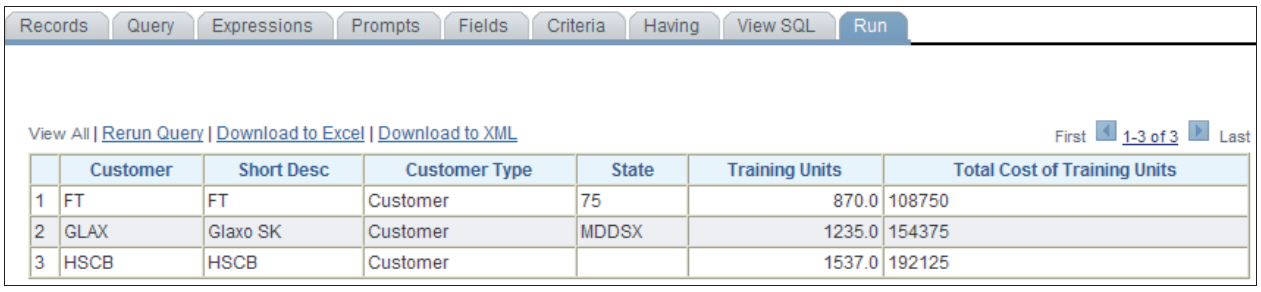
   The Edit Criteria Properties page reappears and displays the expression in the Expression 1 section.

   In this example, the Training Units expression multiplies the value in the TRAINING_UNITS field by 125 to determine the total cost of training units for each customer; the expression serves as selection criteria to display only the customers whose total training unit cost is greater than 60,000 USD:



4. Save the query, and view the results.

   The following query shows the customers whose total training unit cost is greater than 60,000 USD:

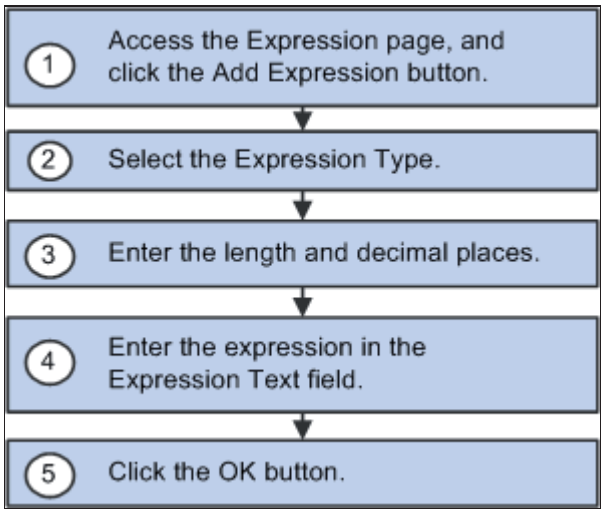


	Customer	Short Desc	Customer Type	State	Training Units	Total Cost of Training Units
1	FT	FT	Customer	75	870.0	108750
2	GLAX	Glaxo SK	Customer	MDDSX	1235.0	154375
3	HSCB	HSCB	Customer		1537.0	192125

Creating Simple Expressions

Steps Used to Create Simple Expressions

Use the following steps to create an expression:



Slide 131

Student Notes

Steps Used to Create Expressions

Use the following steps to create an expression:

1. In Query Manager, select the Expressions tab, and click the Add Expression button.

   Note. To get the results as in the following examples, create a new query using the PSU_ITEM_TBL record and the ITEM_CD field.

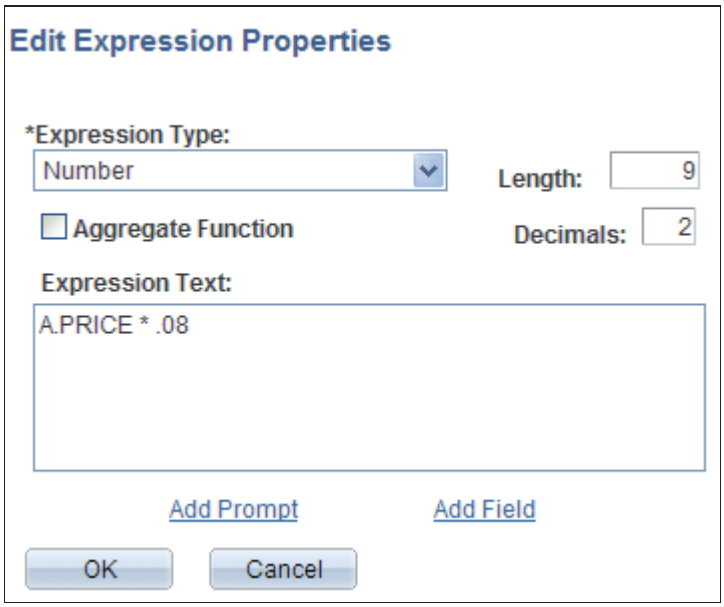
2. On the Edit Expression Properties page, enter the expression properties such as expression type, length, number of decimal place, and aggregate function.

3. Optionally, click the Add Prompt or Add Field link to select a prompt or a field as part of the expression.

   An appropriate page appears enabling you to select a prompt or a field from any record in the query.

4. Complete the calculation in the Expression Text box.

   This example shows the Expression Text box containing a field and other text to perform a calculation:



5. Click the OK button, and confirm that the expression is listed on the Expressions page.

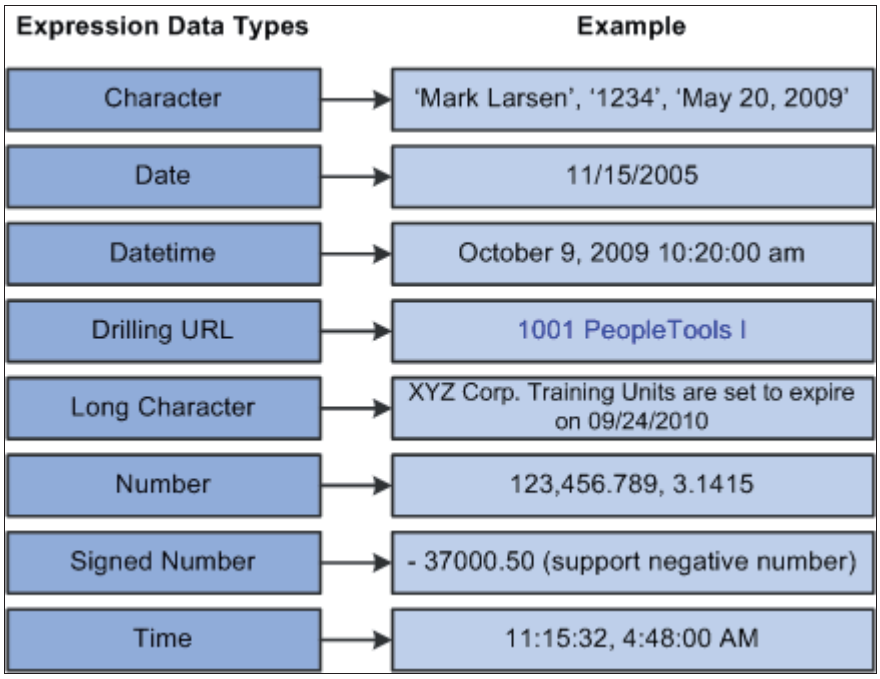
   In this example, the expression A.PRICE * .08 is listed:

6. Save the query.

Creating Simple Expressions (continued)

Expression Data Types

A data type is a categorization of the data that appears in a field or formula. All data that you use in a formula is one of these data types:



Slide 132

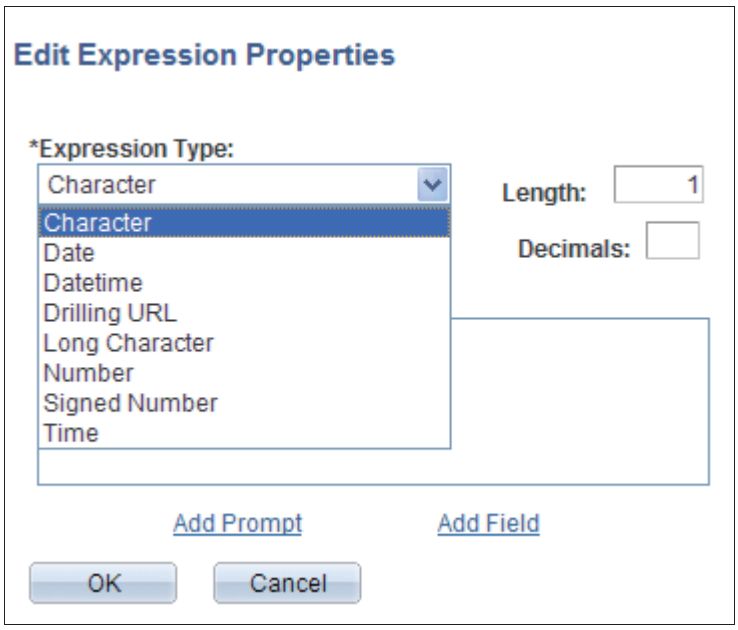
Student Notes

Importance of Data Types

You must be familiar with data types when you work with formulas. Most formulas require and work with only certain data types. For example, you can add two numbers and you can concatenate two strings, but you cannot mix the two data types when adding or concatenating.

Example: Expression Data Types

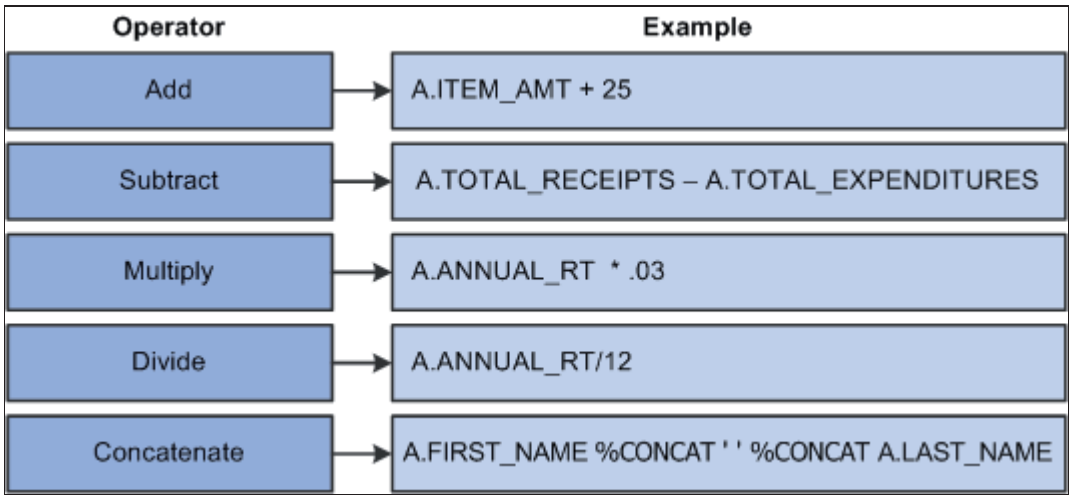
This example shows the Edit Expression Properties page with the list of expression data types:



Creating Simple Expressions (continued)

Operators

This illustrate lists the operators and their examples:



Slide 133

Student Notes

Operators

Operators are symbols, which represent operations and describe an action that takes place between two or more values.

Operators vary based on the formula that you write.

In the following example, the operators concatenate text and field values:

```
'Revenue for ' %CONCAT A.DESCR %CONCAT ' (' %CONCAT A.DEPTID %CONCAT ')'
```

This formula might display this text: *Revenue for Manufacturing (KM003)* where *Manufacturing* is the value in the A.DESCR field and *KM003* is the value in the A.DEPTID field.

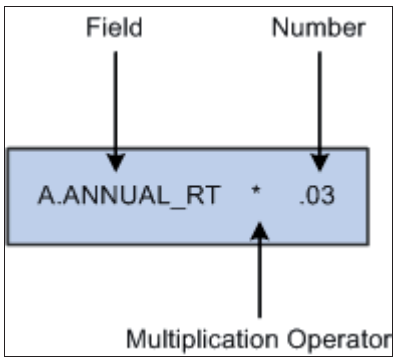
The following example shows the operator that performs mathematical calculations:

```
A.ANNUAL_RT * .03
```

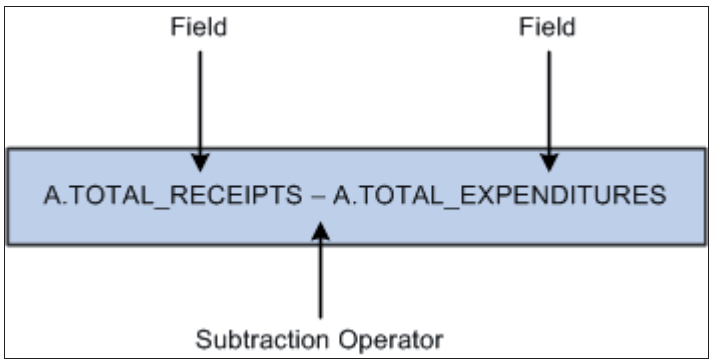
This second formula might display this value: *2700* where *90000* is the value that is in the A.ANNUAL_RT field.

Examples of Fields, Text, and Numbers in Formulas

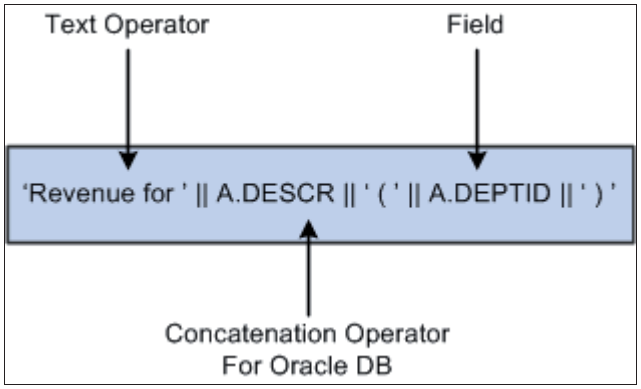
This example shows how to use field, number, and operators to find there percent bonus amount based on annual salary:



This example shows how to use fields and operators to *subtract* total expenditures from total receipts and display the net income:



This example shows how to use fields and operators to display the revenue and departments in one column:



Activity 18: Creating Simple Expressions

In this activity, you will review the activity overview and:

1. Create a query.
2. Create a simple expression.
3. Edit the expression column heading.

Slide 134

Activity Overview

Create a query named CUSTOMER_TRN_UNITS that uses the Customer table (PSU_CUST_TBL) and the following fields and properties:

- CUSTOMER_ID, *Order By 1.*
- DESCR, *Name.*
- CUSTOMER_TYPE, *RFT Long, XLAT Short.*
- CONTRACT_DATE, *RFT Long.*
- TRAINING_UNITS, *RFT Long.*

Create an expression to determine the amount that each customer has spent on training units. The expression multiplies the cost of each training unit by the number of units that each customer owns. Each training unit costs 125.00 USD.

Note. Use *PTRPTG* as the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create the query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Create a query that uses the PSU_CUST_TBL record.
3. Select and edit the following fields and properties:

Field	***Field Edit Properties***
CUSTOMER_ID	*Order By 1*
DESCR	*Name*
CUSTOMER_TYPE	*RFT Long XLAT Short*
CONTRACT_DT	*RFT Long*
TRAINING_UNITS	*RFT Long*

4. Save the query as CUSTOMER_TRN_UNITS, and run the query.

Creating Simple Expressions

To create simple expressions:

1. From the Run page, select the Expressions tab.
2. Click the Add Expression button, and enter the following information:

Page Element	***Value or Status***
Expression Type	*Number*
Length	*10*
Decimal	*2*

3. Click the Add Field link, and select the *A.TRAINING_UNITS* field from the list.
4. Enter ** 125* in the Expression Text box after the field name A.TRAINING_UNITS, so that the code reads like this:

```
A.TRAINING_UNITS * 125
```

5. Click the OK button.
6. Save the query, and click the Use as Field link.

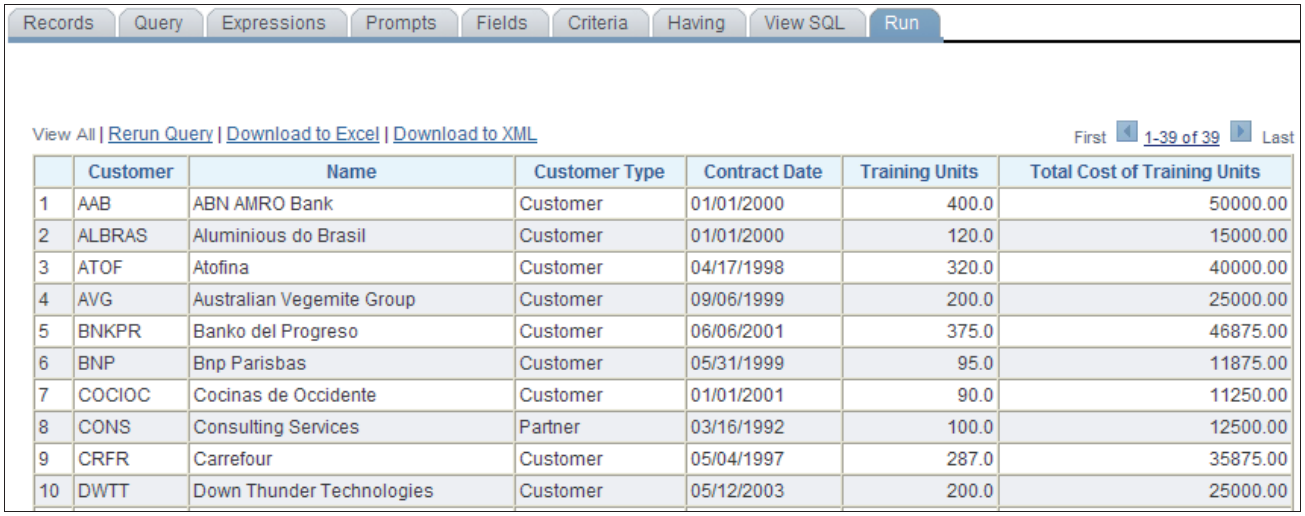
Editing the Expression Column Heading

To edit the expression column heading:

1. On the Fields page, click the Edit button for the expression.
2. Select the Text option, and enter *Total Cost of Training Units* as the heading text.
3. Click the OK button.
4. Save and preview the query.
5. Compare the output with the following results.

Results

This is the returned query results of the CUSTOMER_TRN_UNITS query:



	Customer	Name	Customer Type	Contract Date	Training Units	Total Cost of Training Units
1	AAB	ABN AMRO Bank	Customer	01/01/2000	400.0	50000.00
2	ALBRAS	Aluminious do Brasil	Customer	01/01/2000	120.0	15000.00
3	ATOF	Atofina	Customer	04/17/1998	320.0	40000.00
4	AVG	Australian Vegemite Group	Customer	09/06/1999	200.0	25000.00
5	BNKPR	Banko del Progreso	Customer	06/06/2001	375.0	46875.00
6	BNP	Bnp Parisbas	Customer	05/31/1999	95.0	11875.00
7	COCIOC	Cocinas de Occidente	Customer	01/01/2001	90.0	11250.00
8	CONS	Consulting Services	Partner	03/16/1992	100.0	12500.00
9	CRFR	Carrefour	Customer	05/04/1997	287.0	35875.00
10	DWTT	Down Thunder Technologies	Customer	05/12/2003	200.0	25000.00

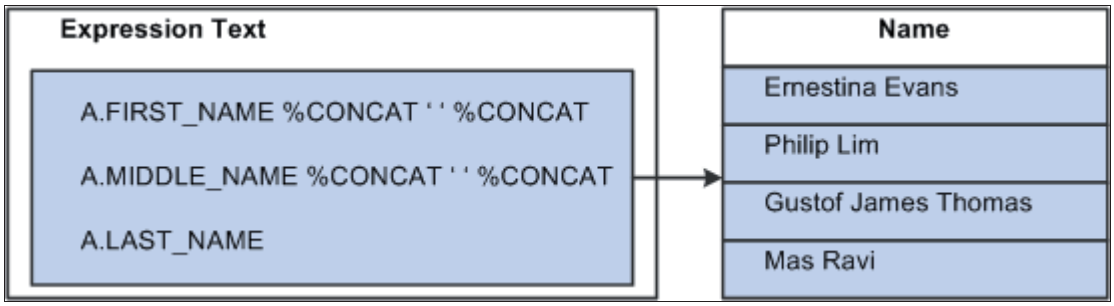
This concludes the activity. Please do not continue.

Using Literals in Expressions

Including Literals in Expressions

Literals are any text values that appear *literally* in the expression. They are useful for combining text from two columns in a query.

This is an example of using the dash and the space as literals in expressions:



Slide 135

Student Notes

Page Used to Manage Literals in Expressions

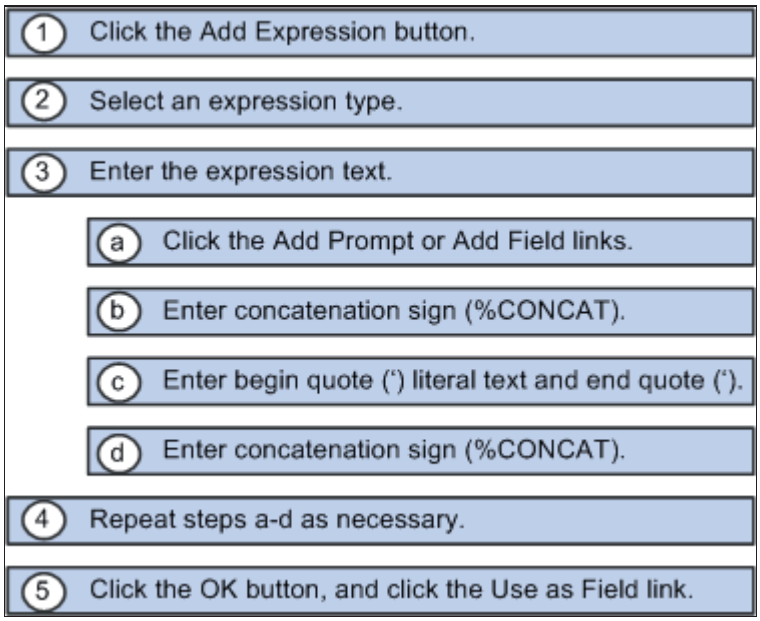
Use this page to manage literals in expressions:

Page Name	*Navigation*
Edit Expression Properties	1. Select Reporting Tools, Query, Query Manager. 2. Create a new query or open an existing one. 3. Select the Expressions tab, and click the appropriate Add Expression or Edit button.

Using Literals in Expressions (continued)

Steps for Using Literals in an Expression

This diagram shows the steps used to add literals in a concatenation expression:

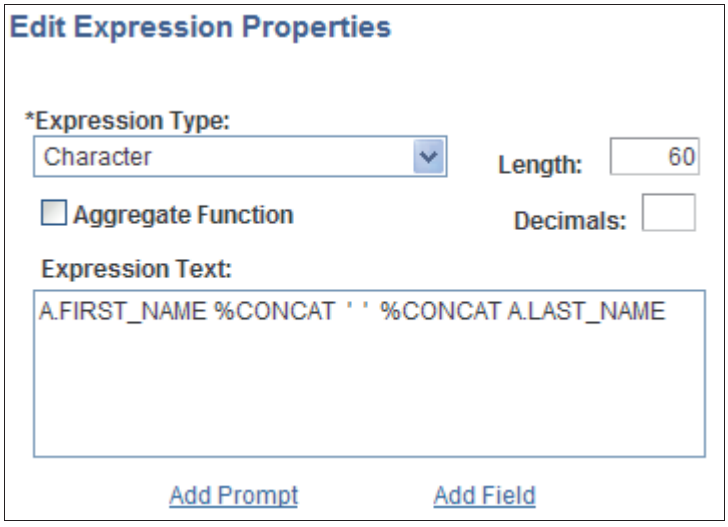


Slide 136

Student Notes

Using Literals in an Expression

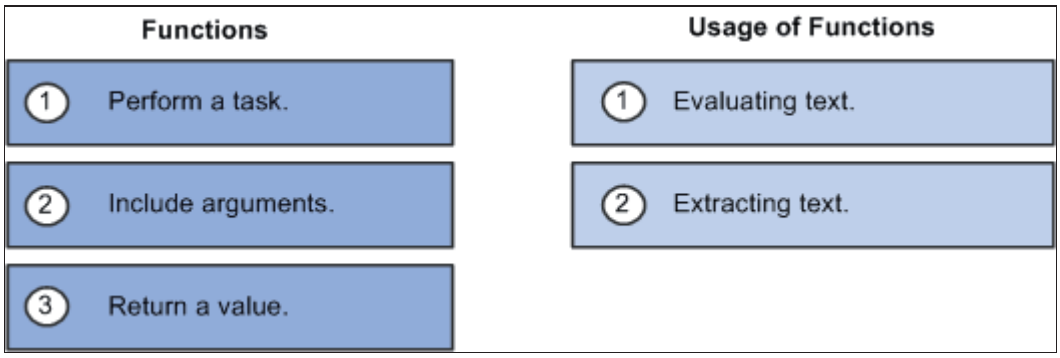
You enclose literals in single quotes (' ') as shown in this example:



Using Functions in Expressions

Functions

This illustrate shows the actions and usages of functions:



Slide 137

Student Notes

Functions

Functions are prebuilt instructions that perform a task. When you include a function in a formula, the program performs the set of operations that are associated with that function so that you don't need to specify each operation separately.

Many functions require you to enter arguments so that the function can perform its task.

Character Functions Returning Number Value

Function: Length(str)

The `Length(str)` function returns the number of characters in the text string that you place in the parentheses.

Note. You must enclose text strings in single quotation marks (' ').
Blank spaces are parts of the character count.

This table provides examples of using the `Length(str)` function:

Example	*Return*
`LENGTH('data')`	4
`LENGTH(' d a t a ')`	9
`LENGTH(A.SSN)`	9

Function: INSTR(str1, str2)

The `INSTR(str1, str2 )` function searches string 1 for string 2. This function reads from the left to the right and returns a number that represents the position of string 2 inside of string 1.

This table provides an example of using the `InStr(str1, str2)` function:

Example	*Return*
`INSTR('Smith', 'm')`	2

Character Functions Returning Character Values

Function: SUBSTR(char, position, substring_length)

The `SUBSTR(char, position, substring_length)` function returns a portion of `char`, beginning at character `position`, `substring_length` characters long.

This table shows an example of using the `SUBSTR(char, position, substring_length)` function to return specified substrings of *ABCDEFG*:

Example	*Return*
`SUBSTR('ABCDEFG' , 3, 4)`	CDEF

Function: INITCAP(char)

The `INITCAP(char)` function returns `char` with the first letter of each word in uppercase and all other letters in lowercase.

This table shows an example of using the `INITCAP(char)` function:

Example	*Return*
`INITCAP ('ORACLE CORPORATION')`	Oracle Corporation

Function: RTRIM(char [, set])

The `RTRIM(char [, set ])` function is useful for formatting the output of a query. It removes all characters that appear in set from the right end of char.

This table shows an example of using the `RTRIM(char [, set ])` function to trim all the right-most occurrences of period, slash, and equal sign from a string:

Example	*Return*
`RTRIM('BASEBALL ')`	BASEBALL

Function: LTRIM(char [, set])

The `LTRIM(char [, set ])` function removes all characters contained in `set` from the left end of char.

Note. If you do not specify `set`, then the default `set` value is a single blank.
If `char` is a character literal, then you must enclose `char` in single quotation marks.

This table shows an example of using the `LTRIM(char [, set ])` function to trim the first redundant word from a group of product names in the products table:

Example	*Return*
`LTRIM('Monitor 17/HR', 'Monitor ')`	17/HR

Combining Functions

You can combine functions to extract information from fields.

In this example, the PERSONAL_DATA.NAME field contains names in the *Lastname,Firstname* format. If you want to show the name in the *FirstName LastName* format, you must display everything, except the comma:

Example	*Return*
A.NAME	ABADIE,Laurence
SUBSTR(A.NAME, INSTR(A.NAME, ',')+1) %CONCAT ' ' %CONCAT SUBSTR(A.NAME, 1, INSTR(A.NAME, ',')-1)	Laurence ABADIE
INITCAP(SUBSTR(A.NAME, INSTR(A.NAME, ',')+1) %CONCAT ' ' %CONCAT SUBSTR(A.NAME, 1, INSTR(A.NAME, ',')-1))	Laurence Abadie

The inner function is executed first. The INSTR() function searches for the first occurrence of a comma in *ABADIE,Laurence* from left to right. After it finds the comma in position 7, it adds *+1* to avoid showing the comma. The SUBSTRING() function only shows the string in the A.NAME field starting in position 8. Concatenation is added between the first name and the last name to include a space. The result is *Laurence* .

```
SUBSTR(A.NAME, INSTR(A.NAME, ',')+1) %CONCAT ' ' %CONCAT
```

To obtain the last name, the SUBSTRING() function starts in position 1 for a length of 6. The result is *ABADIE.*

```
SUBSTR(A.NAME, 1, INSTR(A.NAME, ',')-1)
```

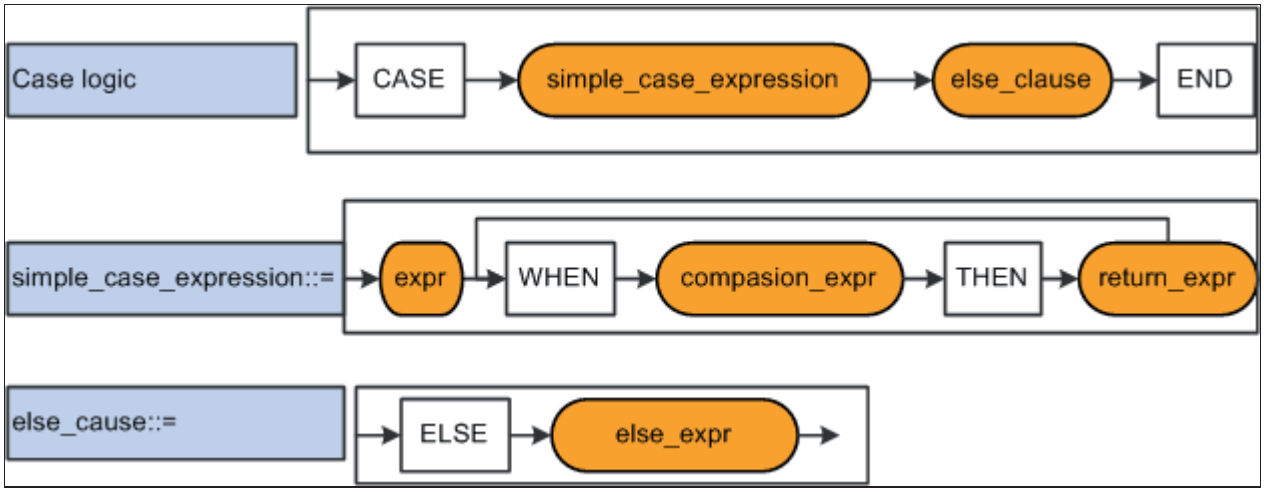
Lastly, to make sure that the full name of *Laurence ABADIE* is in proper case, we use the INITCAP() function. The result is *Laurence Abadie.*

Using Functions in Expressions (continued)

Control Structures: CASE Logic

CASE logic is one of the most useful structures in formulas. It enables you to evaluate the current state of an object, and then perform a calculation or execute a function based on that state.

This diagram illustrates the process flows for CASE logic, `simple_case_expression::=` structure, and `else cause::=` structure:



Slide 138

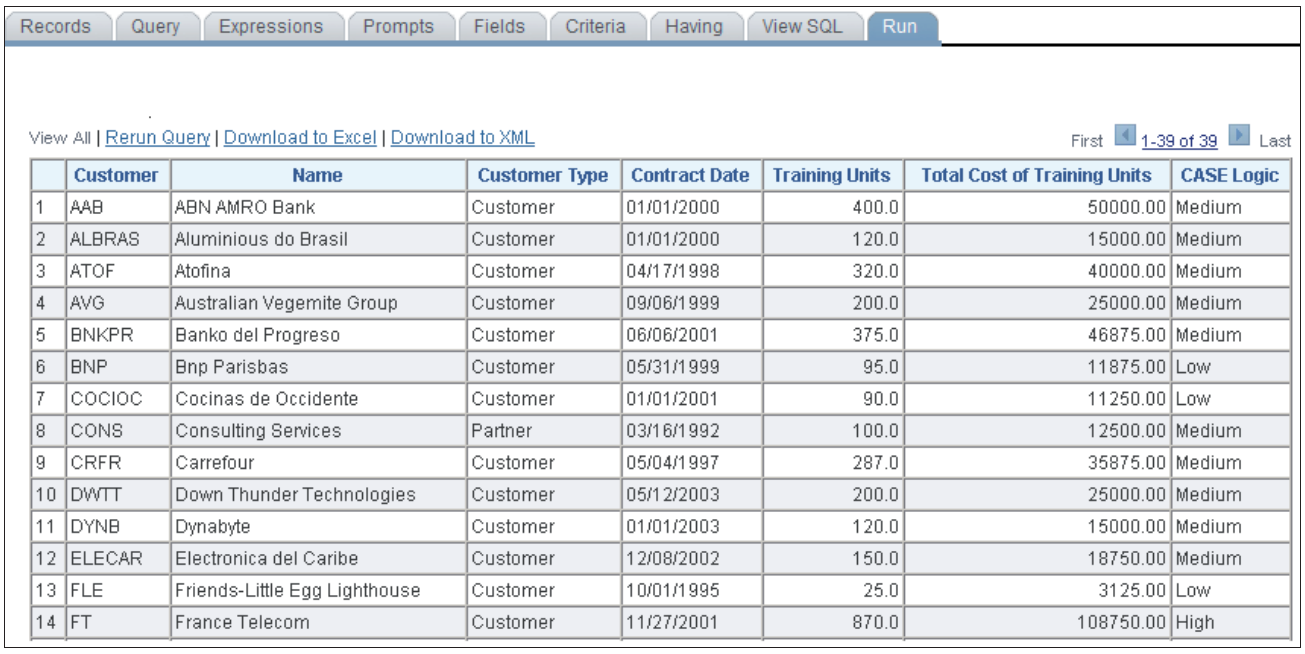
Student Notes

Example: CASE Logic

For each customer in the CUSTOMERS table, the following statement lists the credit limit as *Low* if it equals $100, as *High* if it equals $5000, and as *Medium* if it equals other amounts:

```
CASE
WHEN A.TRAINING_UNITS < 100 THEN 'Low'
WHEN A.TRAINING_UNITS BETWEEN 100 AND 400 THEN 'Medium'
WHEN A.TRAINING_UNITS > 400 THEN 'High'
ELSE 'Unknown'
END
```

This example shows the results of applying CASE logic in the CUSTOMERS table:



	Customer	Name	Customer Type	Contract Date	Training Units	Total Cost of Training Units	CASE Logic
1	AAB	ABN AMRO Bank	Customer	01/01/2000	400.0	50000.00	Medium
2	ALBRAS	Aluminious do Brasil	Customer	01/01/2000	120.0	15000.00	Medium
3	ATOF	Atofina	Customer	04/17/1998	320.0	40000.00	Medium
4	AVG	Australian Vegemite Group	Customer	09/06/1999	200.0	25000.00	Medium
5	BNKPR	Banko del Progreso	Customer	06/06/2001	375.0	46875.00	Medium
6	BNP	Bnp Parisbas	Customer	05/31/1999	95.0	11875.00	Low
7	COCIOC	Cocinas de Occidente	Customer	01/01/2001	90.0	11250.00	Low
8	CONS	Consulting Services	Partner	03/16/1992	100.0	12500.00	Medium
9	CRFR	Carrefour	Customer	05/04/1997	287.0	35875.00	Medium
10	DWTT	Down Thunder Technologies	Customer	05/12/2003	200.0	25000.00	Medium
11	DYNB	Dynabyte	Customer	01/01/2003	120.0	15000.00	Medium
12	ELECAR	Electronica del Caribe	Customer	12/08/2002	150.0	18750.00	Medium
13	FLE	Friends-Little Egg Lighthouse	Customer	10/01/1995	25.0	3125.00	Low
14	FT	France Telecom	Customer	11/27/2001	870.0	108750.00	High

Activity 19: Using the SUBSTR Function in Expressions

In this activity, you will review the activity overview and:

1. Create a new query.
2. Use the SUBSTR function in an expression.
3. Use an expression as a field.

Slide 139

Activity Overview

Create the *SUBSTR_IN_EXPR* query using the PERSONAL_DATA record and its NAME field.

Add a character expression with the length of 15 and the value text of *INITCAP(SUBSTR(A.NAME, INSTR(A.NAME, ',')+1) || ' ' || SUBSTR(A.NAME, 1, INSTR(A.NAME, ',')-1)).* And then use the expression as a field with the heading text *Full Name.*

Note. Use *PTRPTG* as the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a New Query

To create a new query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. In Query Manager, create a new query using the following information:

Record	***Field***
PERSONAL_DATA	*NAME*

3. Save the query as *SUBSTR_IN_EXPR*.

Using the SUBSTR Function in Expressions

To use the `SUBSTR` function in an expression:

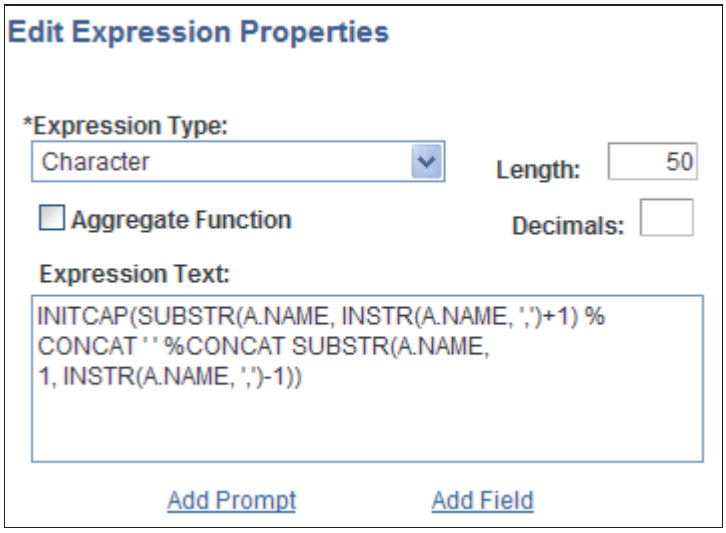
1. Access the Expressions page, and click the Add Expression button.
2. On the Edit Expression Properties page, enter the following information:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*50*
Decimal	<Blank>
Expression Text	*INITCAP(SUBSTR(A.NAME, INSTR(A.NAME, ',')+1) %CONCAT ' '%CONCAT SUBSTR(A.NAME, 1, INSTR(A.NAME, ',')-1))*

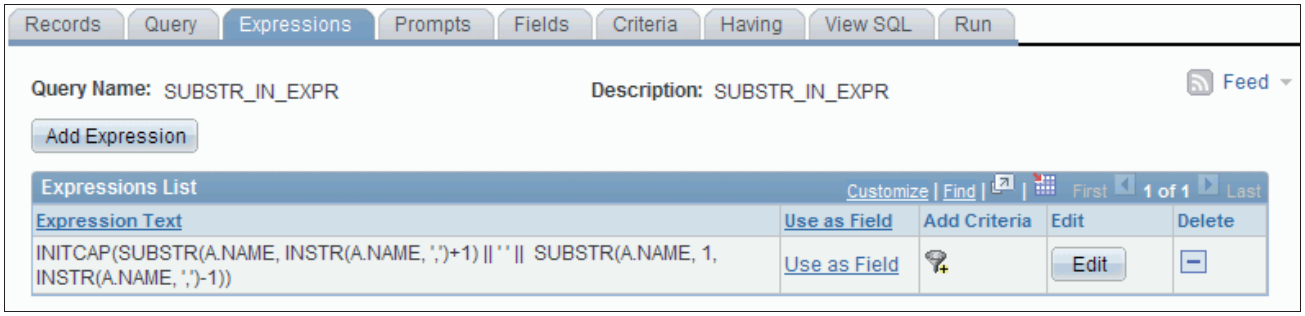
3. Click the OK button, and confirm that the new expression is added into the Expressions page.
4. Compare the output with the following results.

Results

This example shows the expression properties defined. Notice the Expression Text field:



This example shows the Expressions page listing the INITCAP expression:



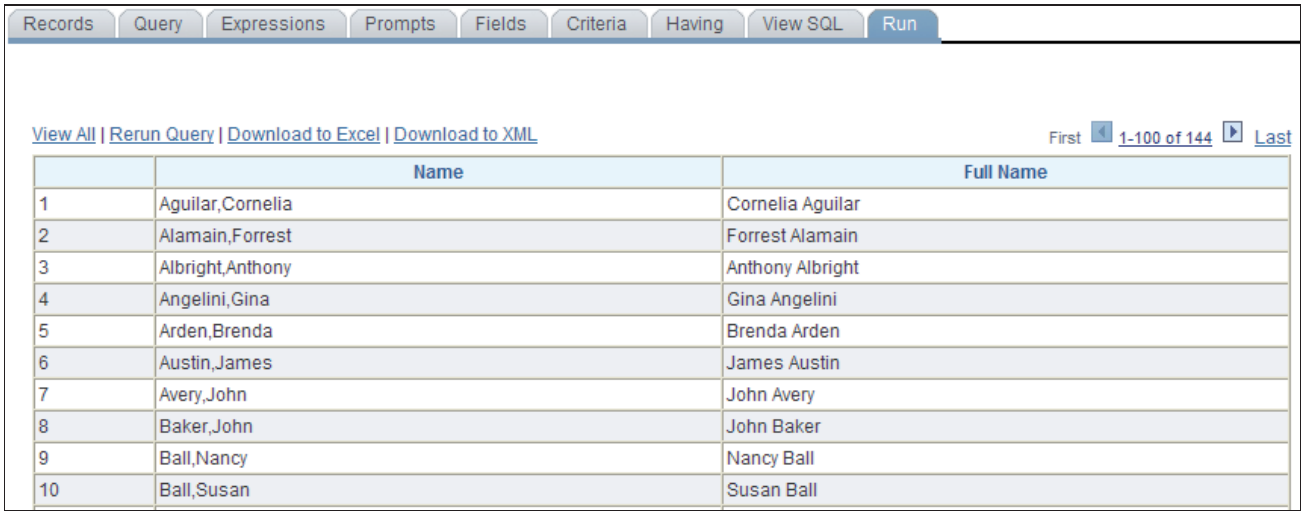
Using Expression as Field

To use an expression as a field:

1. From the Expressions page, click the Use as Field link.
2. In the Fields page, change the heading text of the INITCAP expression to *Full Name.*
3. Save the query, and run the output.
4. Compare the output with the following results.

Results

This example shows the Full Name column that include both the first names and the last names:



	Name	Full Name
1	Aguilar,Cornelia	Cornelia Aguilar
2	Alamain,Forrest	Forrest Alamain
3	Albright,Anthony	Anthony Albright
4	Angelini,Gina	Gina Angelini
5	Arden,Brenda	Brenda Arden
6	Austin,James	James Austin
7	Avery,John	John Avery
8	Baker,John	John Baker
9	Ball,Nancy	Nancy Ball
10	Ball,Susan	Susan Ball

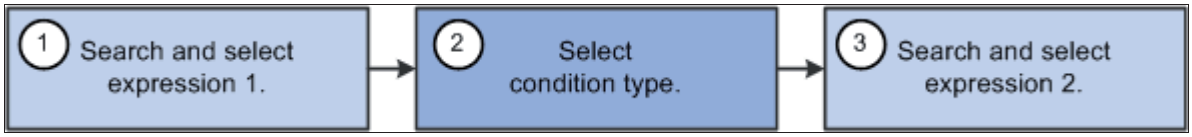
This concludes the activity. Please do not continue.

Using Expressions in Criteria

Applying Expressions to Criteria

You use expressions to refine selection criteria.

This diagram shows the steps used to apply expressions to criteria:



Slide 140

Student Notes

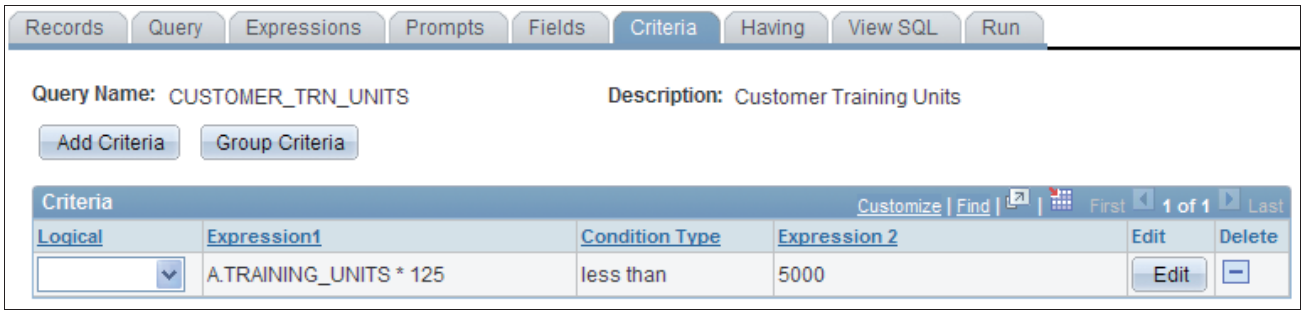
Adding Expressions to Criteria

You add expressions to criteria using the following pages:

- The Expressions page.
- The Criteria page.
- The Fields page (if the expression is used as a field).

Example: Adding Expressions to Criteria Rows

This example Criteria page includes an expression that filters data and displays only customers whose total training unit cost is less than 5000 USD:

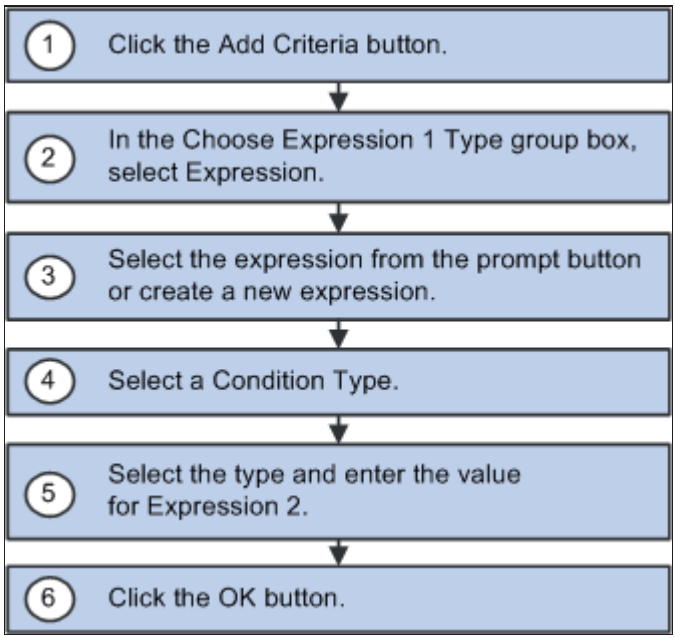


Notice that when you view the results, the row count is reduced considerably.

Using Expressions in Criteria (continued)

Steps for Adding Expressions to Criteria

Use these steps to add an expression to a row of criteria from the Criteria page:



Slide 141

Student Notes

Using Expressions in Criteria

Use these steps to add an expression as criteria on the Criteria page:

1. In the Criteria page, click the Add Criteria button to create a new row of criteria.

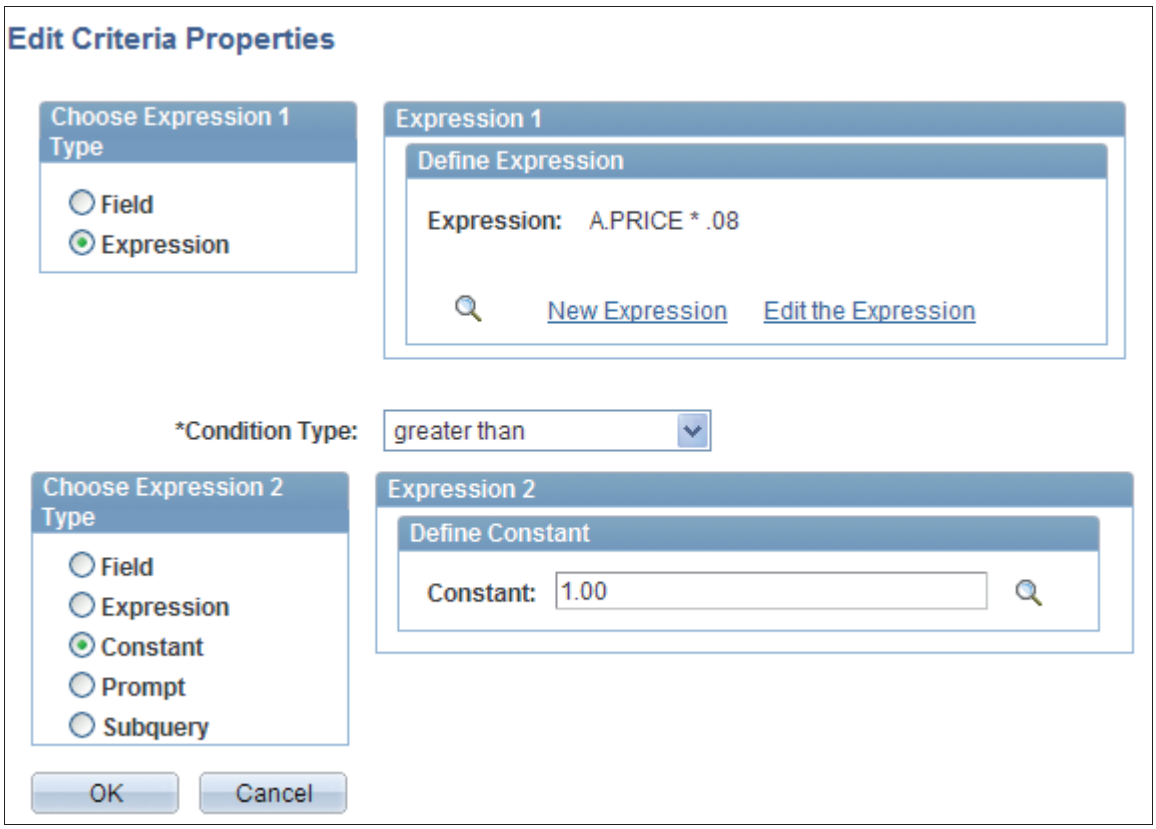
   The Edit Criteria Properties page appears.

2. Select the Expression option in the Expression 1 Type group box.

3. Click the Select Expression lookup icon, and select an existing expression from the expression list of this query.

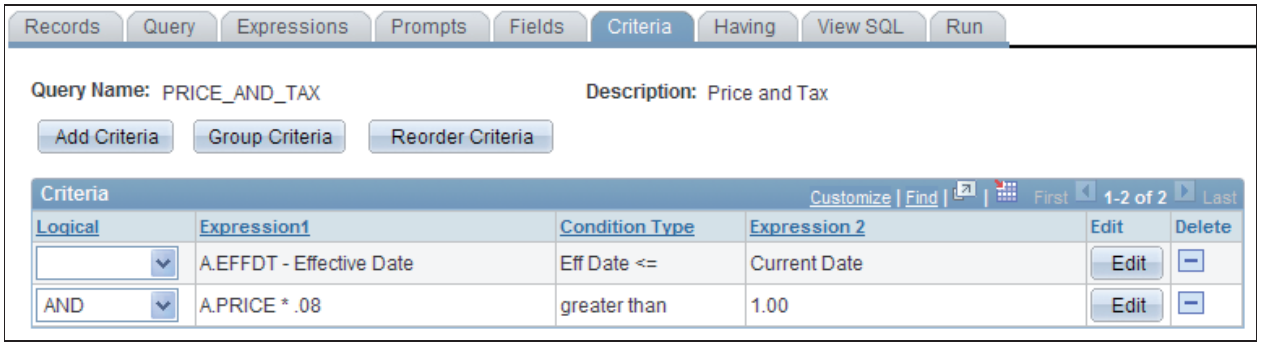
4. Define condition type, expression 2 type, and expression 2.

   This example shows the Edit Criteria Properties after you added the `A.PRICE * .08` expression and other criteria values:



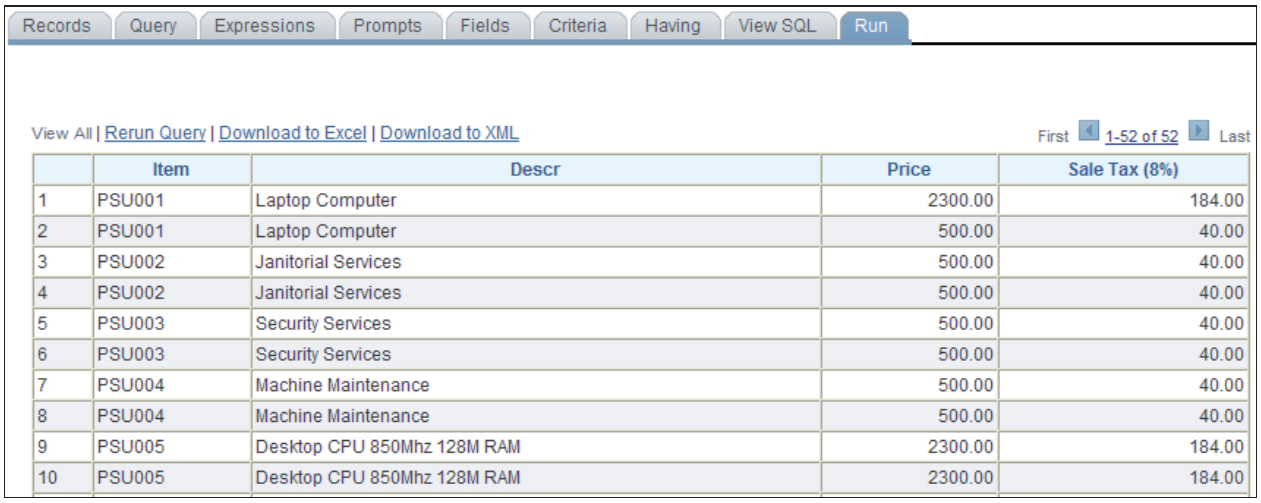
5. Click the OK button, and confirm that the new criteria is listed in the Criteria page.

   This example shows the `A.PRICE * .08` criteria is added:



6. Save and run the query.

   This example displays only the items with a sales tax greater than 1.00:

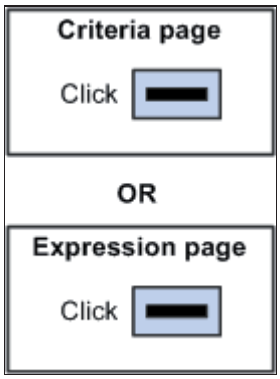


	Item	Descr	Price	Sale Tax (8%)
1	PSU001	Laptop Computer	2300.00	184.00
2	PSU001	Laptop Computer	500.00	40.00
3	PSU002	Janitorial Services	500.00	40.00
4	PSU002	Janitorial Services	500.00	40.00
5	PSU003	Security Services	500.00	40.00
6	PSU003	Security Services	500.00	40.00
7	PSU004	Machine Maintenance	500.00	40.00
8	PSU004	Machine Maintenance	500.00	40.00
9	PSU005	Desktop CPU 850Mhz 128M RAM	2300.00	184.00
10	PSU005	Desktop CPU 850Mhz 128M RAM	2300.00	184.00

Using Expressions in Criteria (continued)

Deleting Expressions

Expressions are deleted from either the Criteria page or the Expressions page as shown in this illustration:

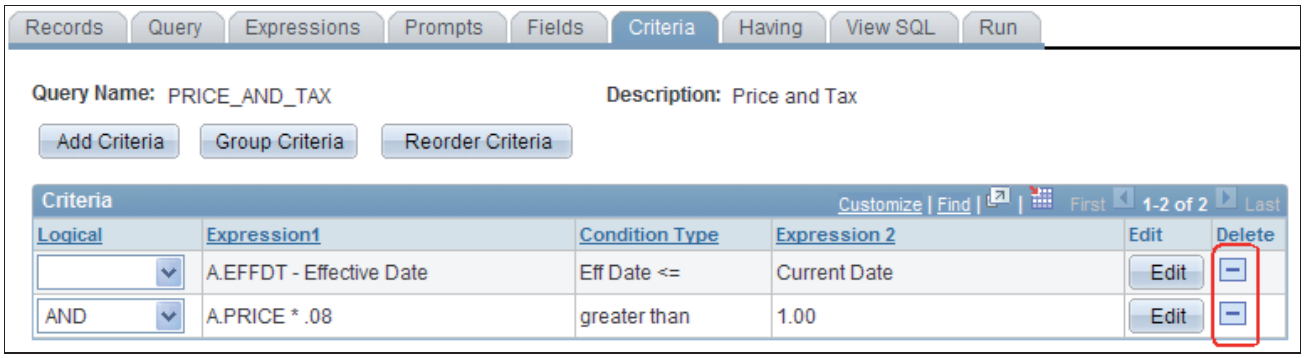


Slide 142

Student Notes

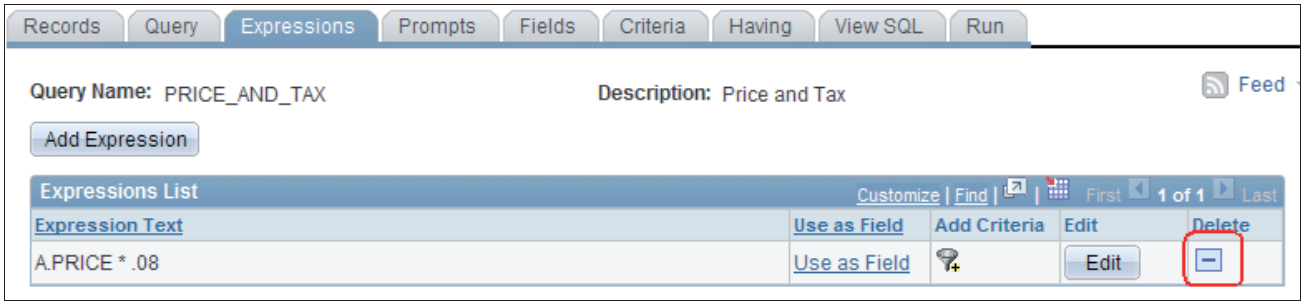
Deleting Expressions from the Criteria Page

To delete a criteria from the Criteria page, click the Delete icon associated with the row of criteria that uses an expression:



Deleting Expressions from the Expressions Page

To delete an expression from the Expressions page, click the Delete button associated with the expression:



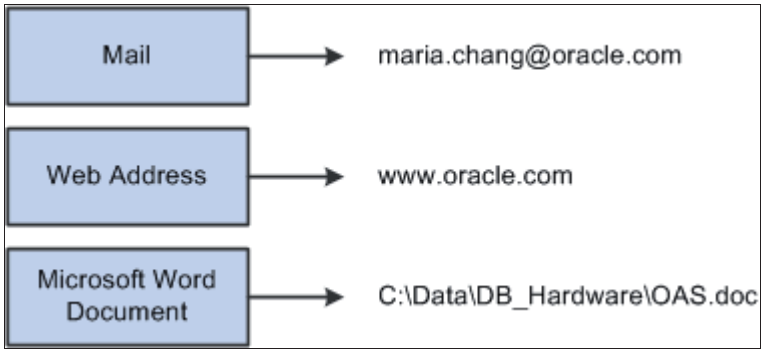
Note. When you delete expressions from the Expressions page, you remove all rows of criteria that reference those expressions.

Using Links in Expressions

Links in Expressions

Query Manager enables you to include links in expressions.

This illustrate shows examples of links that can be used in expressions:



Slide 143

Student Notes

Link Types

Link types include:

- Email links.
- Internet and intranet links.
- Document links.

Commands Used in Links

This table lists seven protocols you use in expressions and the actions they perform:

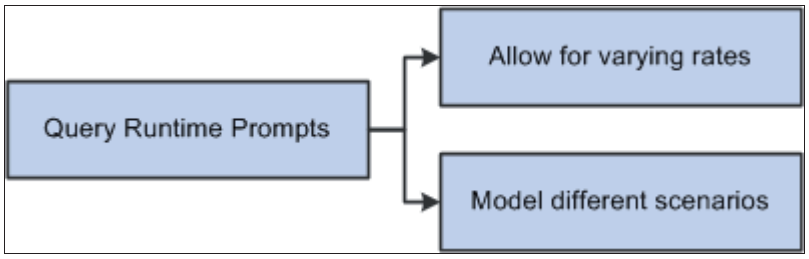
Protocol	***Use***
HTTP:	Access the internet.
HTTPS:	Securely transfer private documents over the internet.
FTP:	Exchange files over the internet.
\\	Access internal network shared drives.
NEWS:	Access news groups.
FILE:	Exchange files using a local computer.
MAILTO:	Access an email application. For example, Lotus Notes or Microsoft Outlook.

Using Prompts in Expressions

Query Runtime Prompts

You add a prompt to expressions to enable users to enter values that become part of the calculation at runtime.

This diagram illustrates two purposes for using prompts in expressions:

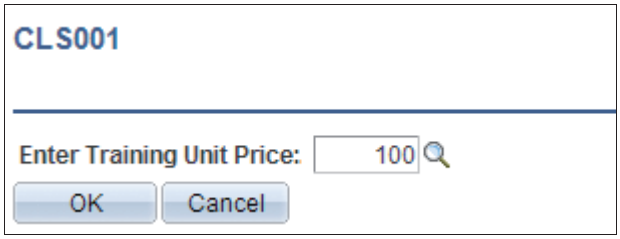


Slide 144

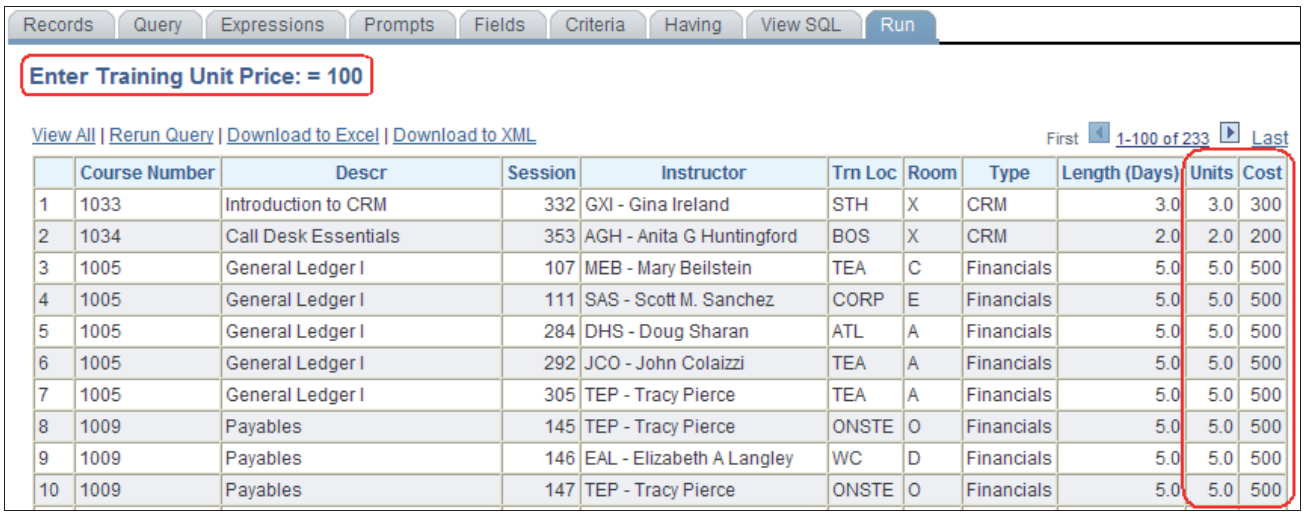
Student Notes

Example: Runtime Prompts in Queries

The training unit cost varies with the number of units the customer purchases. In this example, users enter *100* as cost per training unit at runtime:



In the Run page, this query displays the cost of each course when users enter a Training Unit Cost of *100:*



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

Enter Training Unit Price: = 100

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

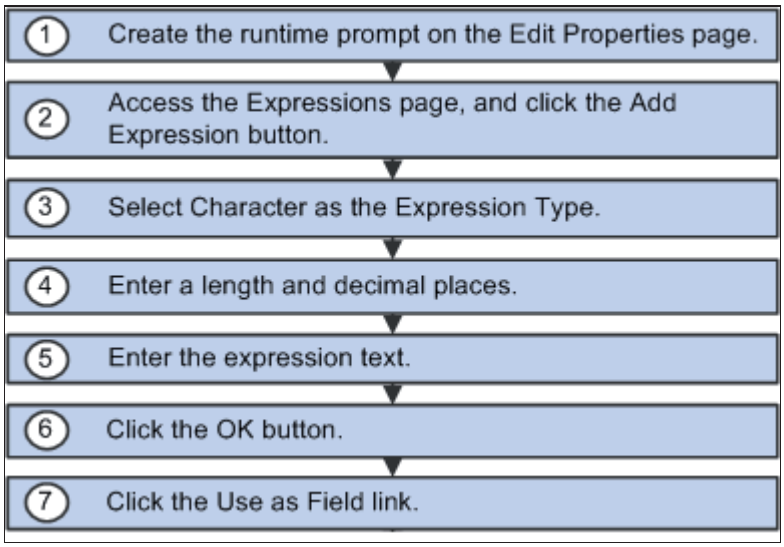
First 1-100 of 233 Last

	Course Number	Descr	Session	Instructor	Trn Loc	Room	Type	Length (Days)	Units	Cost
1	1033	Introduction to CRM	332	GXI - Gina Ireland	STH	X	CRM	3.0	3.0	300
2	1034	Call Desk Essentials	353	AGH - Anita G Huntingford	BOS	X	CRM	2.0	2.0	200
3	1005	General Ledger I	107	MEB - Mary Beilstein	TEA	C	Financials	5.0	5.0	500
4	1005	General Ledger I	111	SAS - Scott M. Sanchez	CORP	E	Financials	5.0	5.0	500
5	1005	General Ledger I	284	DHS - Doug Sharan	ATL	A	Financials	5.0	5.0	500
6	1005	General Ledger I	292	JCO - John Colaizzi	TEA	A	Financials	5.0	5.0	500
7	1005	General Ledger I	305	TEP - Tracy Pierce	TEA	A	Financials	5.0	5.0	500
8	1009	Payables	145	TEP - Tracy Pierce	ONSTE	O	Financials	5.0	5.0	500
9	1009	Payables	146	EAL - Elizabeth A Langley	WC	D	Financials	5.0	5.0	500
10	1009	Payables	147	TEP - Tracy Pierce	ONSTE	O	Financials	5.0	5.0	500

Using Prompts in Expressions (continued)

Steps for Using Prompts in Expressions

This diagram shows the steps used to apply runtime prompts in an expression:



Slide 145

Student Notes

Steps to Create Prompts

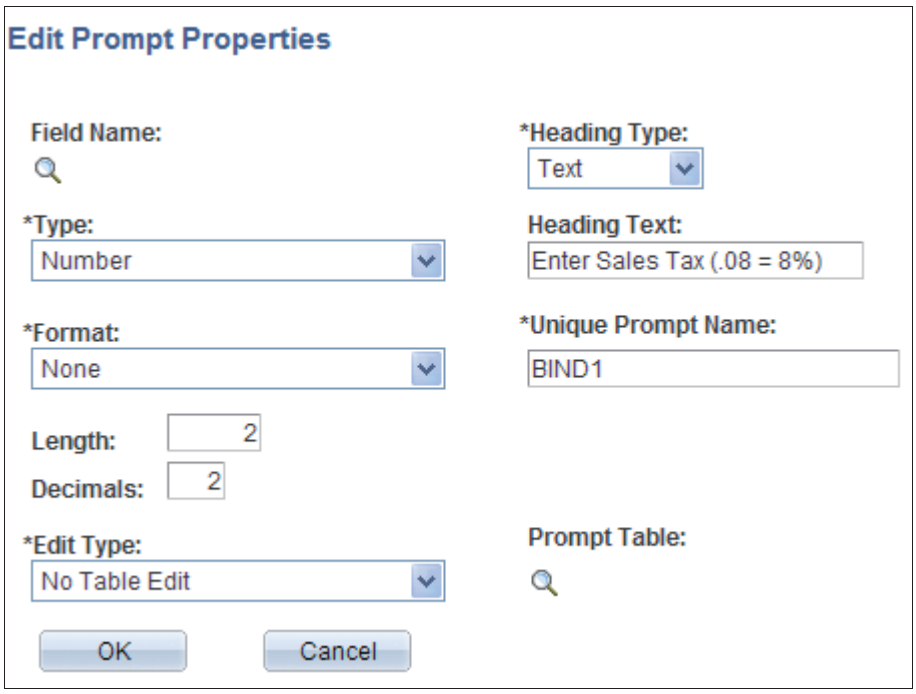
You create the prompt before you create the expression. To create a prompt:

1. In Query Manager, access the Prompts page and click the Add Prompt button.
2. Enter prompt properties.
3. Click the OK button.

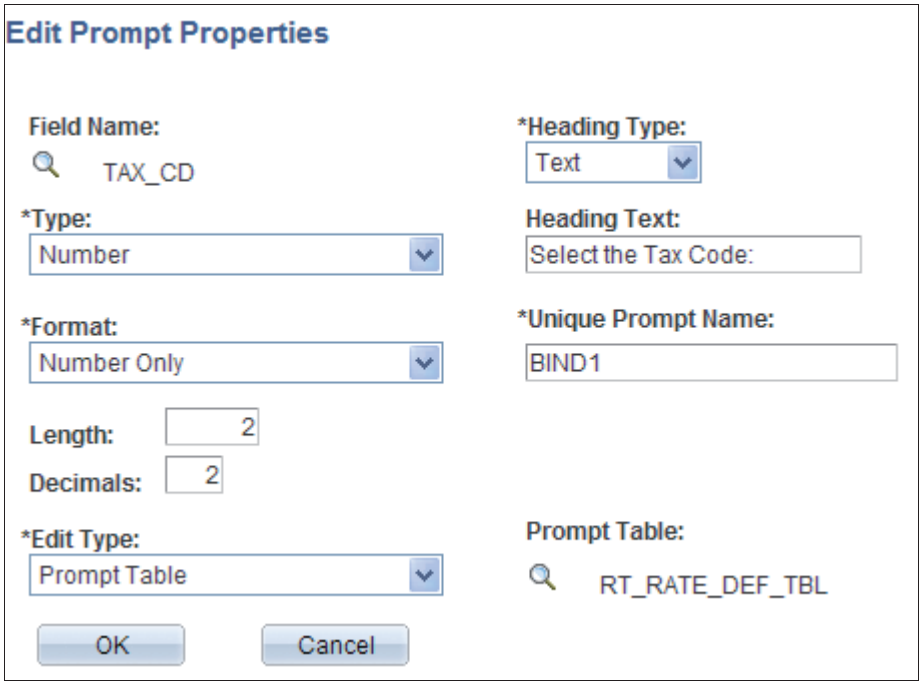
Example: Prompts in Expressions

To enable users to enter any value at the prompt, leave the Field Name blank and select *No Table Edit* as the type. In this case, the database does not validate the prompt value.

This example shows the prompt that enables users to supply the tax rate as a decimal value, but the value is not system validated:



This example shows the prompt that enables users to select only a tax rate in the Real Time Tax Rate table. Notice that you need to search for and select a field name, and then select the edit type to enable users to select a value from a list of predefined values:



Note. Consider the requirements of the expression when you enter the other properties of the prompt field such as type, format, and length. This is important particularly in mathematical calculations.
Consider the query end-user when you enter the heading text that the prompt displays at runtime. It should be concise, but instructive.

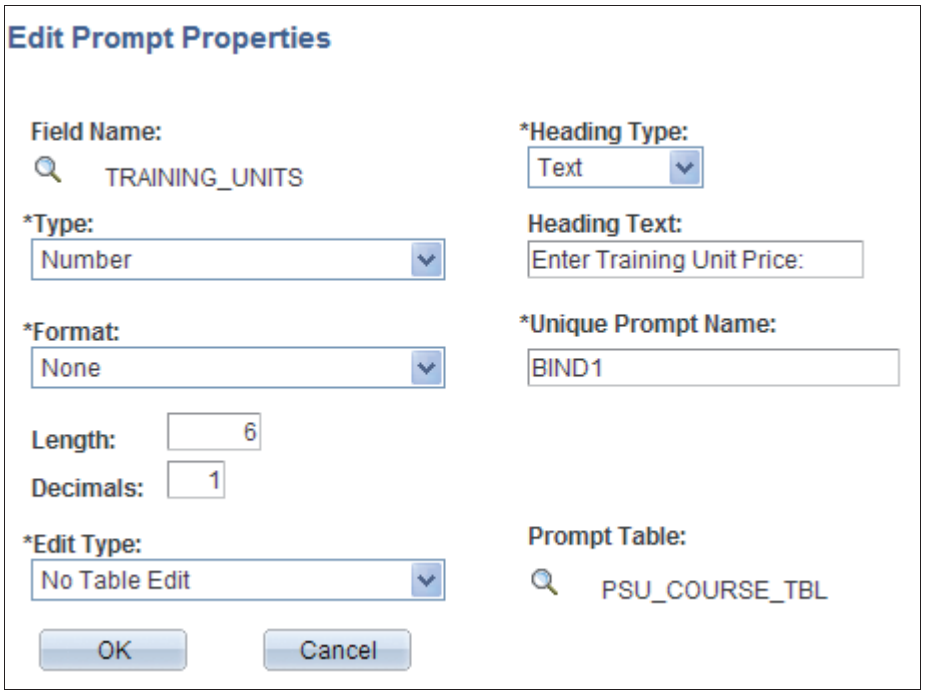
Steps to Use Prompts in Expressions

After creating the prompt, use these steps to apply it in an expression:

1. Select the Expressions tab, and click the Add Expression button.
2. Click the Add Prompt link, and then select a prompt from the list.
3. Complete the expression, and click the OK button.
4. Click the Use as Field link, and verify that the expression is listed on the Fields page.
5. Edit the field properties, if necessary.

Example: Using a Prompt to Calculate Training Unit Cost per Course

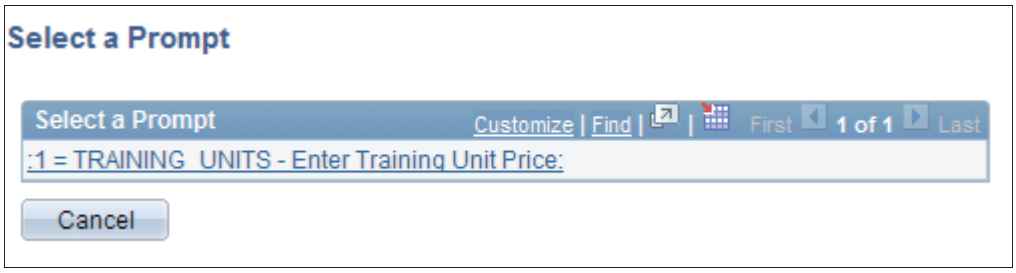
Suppose that you have successfully created the *Enter Training Unit Price* prompt in the CLS001 query, as shown in this example:



To use this *Enter Training Unit Price* prompt to calculate the training unit cost for each course:

1. From the Expressions page, click the Add Expression button to edit the expression properties.
2. Click the Add Prompt link.

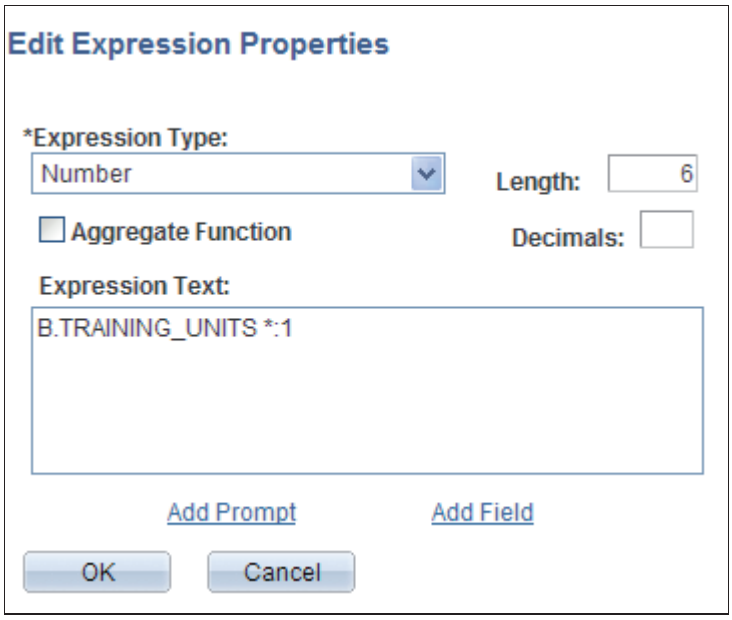
   All prompts that you created in this query appear on the Select a Prompt page, as shown in this example:



3. Click a prompt link and the prompt appears in the expression text as *:<number>*.

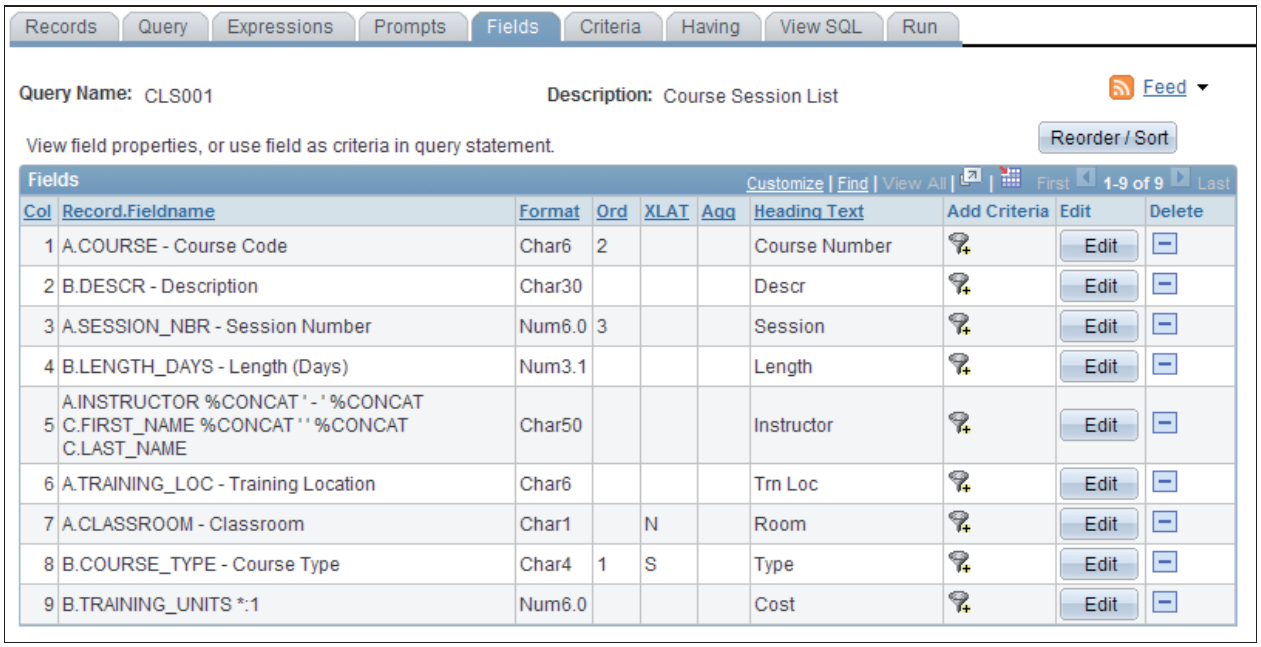
4. Add a field, a calculation, or text to complete the expression.

   This example shows the expression that multiplies the B.TRAINING_UNITS field value by the value the user enters in the prompt, which is indicated by *:1* in this example:



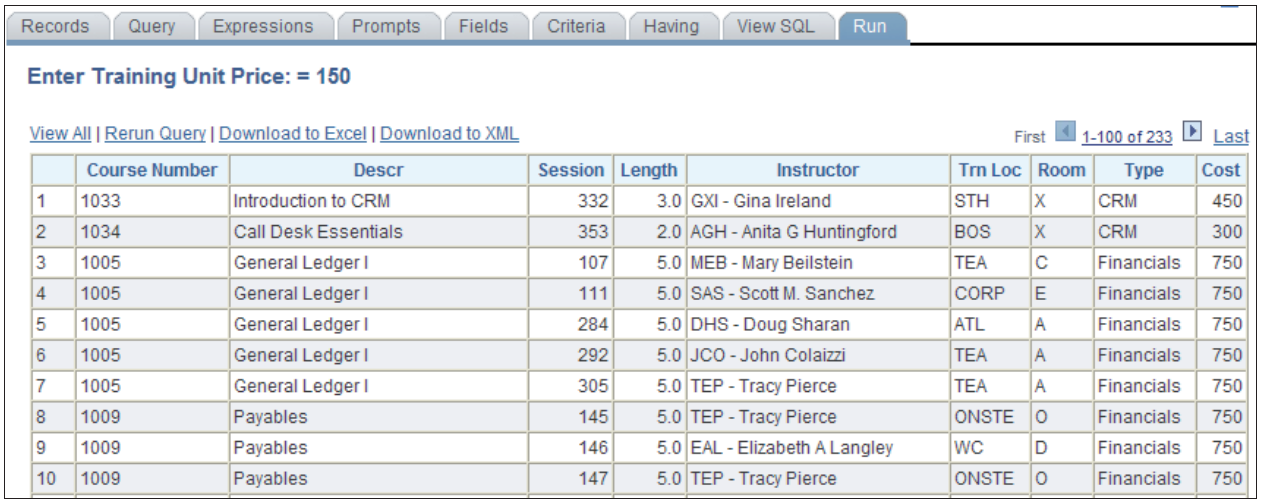
5. Access the Fields page and edit the column heading as needed.

   In this example, fields five and nine are expressions used as fields in this query:



Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete
1	A.COURSE - Course Code	Char6	2			Course Number		Edit	
2	B.DESCR - Description	Char30				Descr		Edit	
3	A.SESSION_NBR - Session Number	Num6.0	3			Session		Edit	
4	B.LENGTH_DAYS - Length (Days)	Num3.1				Length		Edit	
5	A.INSTRUCTOR %CONCAT ' - ' %CONCAT C.FIRST_NAME %CONCAT ' ' %CONCAT C.LAST_NAME	Char50				Instructor		Edit	
6	A.TRAINING_LOC - Training Location	Char6				Trn Loc		Edit	
7	A.CLASSROOM - Classroom	Char1		N		Room		Edit	
8	B.COURSE_TYPE - Course Type	Char4	1	S		Type		Edit	
9	B.TRAINING_UNITS *:1	Num6.0				Cost		Edit	

6. Run the query and examine the output.

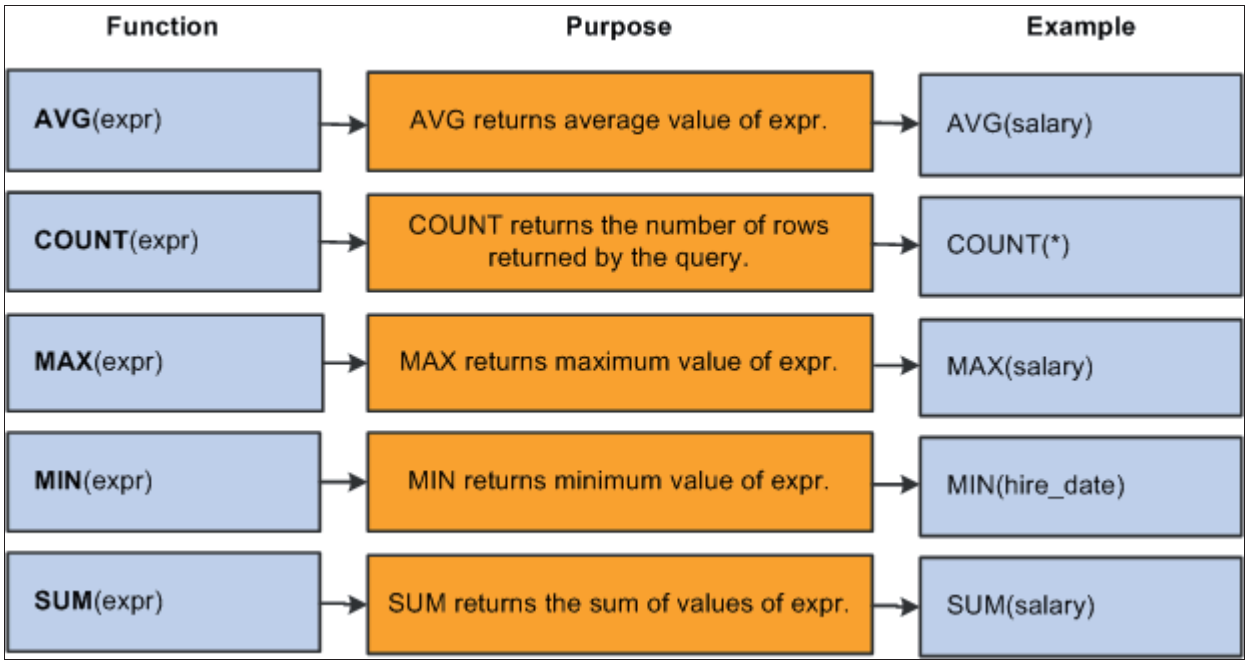


	Course Number	Descr	Session	Length	Instructor	Trn Loc	Room	Type	Cost
1	1033	Introduction to CRM	332	3.0	GXI - Gina Ireland	STH	X	CRM	450
2	1034	Call Desk Essentials	353	2.0	AGH - Anita G Huntingford	BOS	X	CRM	300
3	1005	General Ledger I	107	5.0	MEB - Mary Beilstein	TEA	C	Financials	750
4	1005	General Ledger I	111	5.0	SAS - Scott M. Sanchez	CORP	E	Financials	750
5	1005	General Ledger I	284	5.0	DHS - Doug Sharan	ATL	A	Financials	750
6	1005	General Ledger I	292	5.0	JCO - John Colaizzi	TEA	A	Financials	750
7	1005	General Ledger I	305	5.0	TEP - Tracy Pierce	TEA	A	Financials	750
8	1009	Payables	145	5.0	TEP - Tracy Pierce	ONSTE	O	Financials	750
9	1009	Payables	146	5.0	EAL - Elizabeth A Langley	WC	D	Financials	750
10	1009	Payables	147	5.0	TEP - Tracy Pierce	ONSTE	O	Financials	750

Using Aggregates in Expressions

Combining Aggregates with Expressions

Query Manager enables you to use predefined aggregate functions with expressions. Combine simple operations with aggregate function to create complex calculations. This diagram lists the aggregate functions, purposes, and examples:

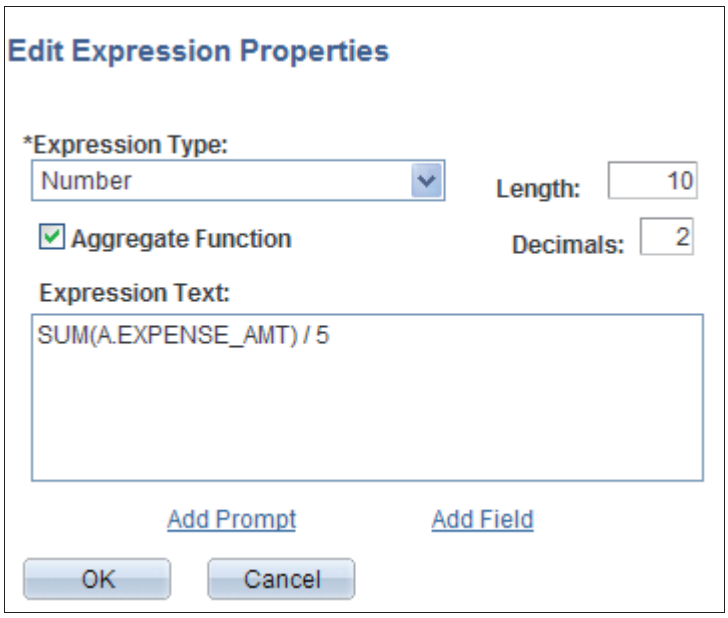


Slide 146

Student Notes

Example: Average Aggregate

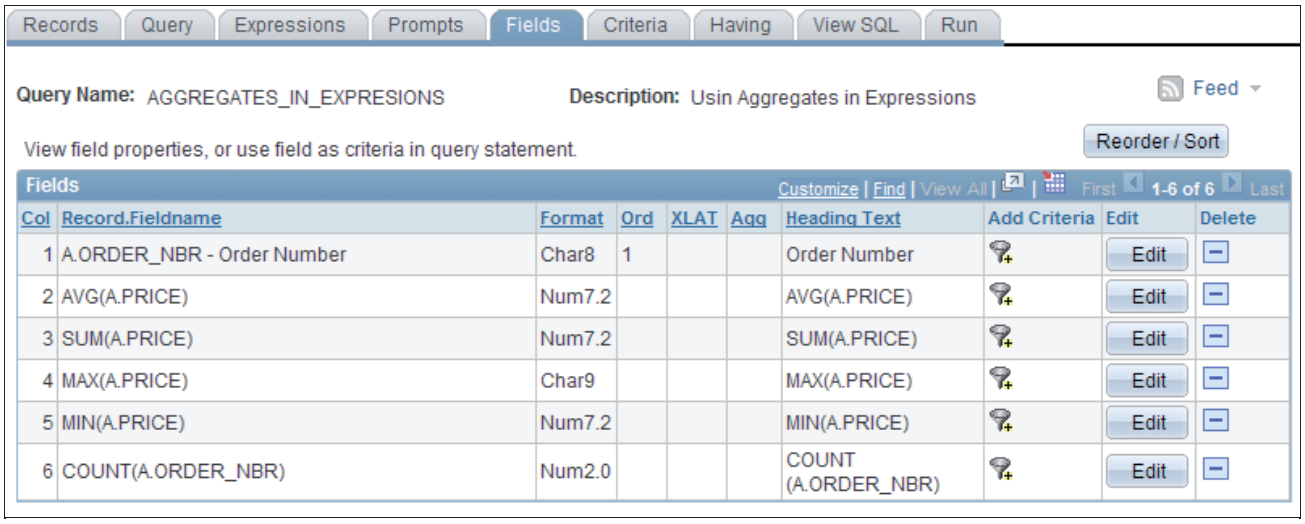
The expression in this example shows the total of all expense amount divided by five days to display the daily employee expenses:



Using Multiple Aggregates with Expressions

PeopleSoft Query enables you to combine multiple aggregates in the same query.

This example shows the Fields page with multiple aggregate expressions used as fields:

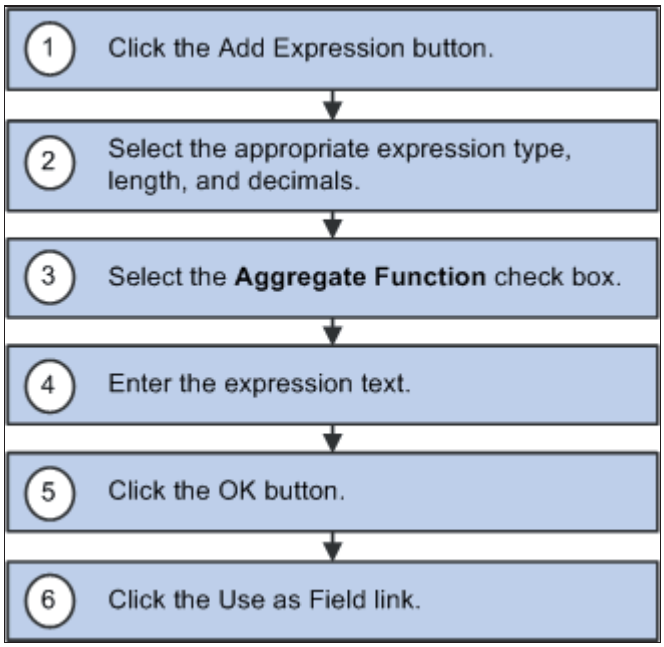


Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete
1	A.ORDER_NBR - Order Number	Char8	1			Order Number		Edit	
2	AVG(A.PRICE)	Num7.2				AVG(A.PRICE)		Edit	
3	SUM(A.PRICE)	Num7.2				SUM(A.PRICE)		Edit	
4	MAX(A.PRICE)	Char9				MAX(A.PRICE)		Edit	
5	MIN(A.PRICE)	Num7.2				MIN(A.PRICE)		Edit	
6	COUNT(A.ORDER_NBR)	Num2.0				COUNT (A.ORDER_NBR)		Edit	

Using Aggregates in Expressions (continued)

Steps for Using an Aggregate in an Expression

This diagram shows the steps used to add an aggregate in an expression:



Slide 147

Student Notes

Steps Used to Add Aggregates to Expressions

To add aggregate functions to expressions:

1. Access Query Manager, and add or open an existing query.
2. Access the Expressions page, and click the Add Expression button.
3. On the Edit Expression Properties page, enter or select the values for expression type, length, and decimals.
4. Select the Aggregate Function check box.
5. Enter the function in the Expression Text field.

Note. Using correct syntax, you must manually enter the aggregate function and any other required parameters.

---

6. Click the OK button to return to the Expressions page.
7. Click the Use as Field button associated with the aggregate that you just created.

8. Optionally, repeat Step 2 through Steps 7 to add additional aggregate functions.

   For example, you can add AVG, SUM, MAX, MIN, or COUNT aggregates to the expression list.

Activity 20: Using Aggregate Functions in Expressions

In this activity, you will review the activity overview and:

1. Create a new query.
2. Add the AVG aggregate function to an expression.
3. Add the COUNT aggregate function to an expression.
4. Add the MAX aggregate function to an expression.
5. Add the MIN aggregate function to an expression.
6. Add the SUM aggregate function to an expression.
7. Edit fields and preview the results.

Slide 148

Activity Overview

In Query Manager, create a new query named *AGGREGATES_IN_EXPR* using the ORD_DTL record and the *ORDER_NBR* field.

Add multiple aggregate functions to count the order numbers; and to display the average, total number, maximum, and minimum of prices.

Use all aggregate expressions as fields with descriptive heading texts, and order the results by the ORDER_NBR field.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a New Query

To create a new query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. In Query Manager, create a new query using the following information:

Record	***Field***
ORD_DTL	*ORDER_NBR*

3. Save the query as *AGGREGATES_IN_EXPR.*

Adding the AVG Aggregate Function to an Expression

To add the AVG aggregate function to an expression:

1. In the Expressions page, click the Add Expression button.
2. Enter the following information for the AVG aggregate:

Page Element	***Value or Status***
Expression Type	*Number*
Aggregate Function	Selected
Length	*9*
Decimals	*2*
Expression Text	AVG(A.PRICE)

3. Click the OK button, and click the Use as Field link for the AVG(A.PRICE) expression.
4. Save the query.

Adding the COUNT Aggregate Function to an Expression

To add the COUNT aggregate function to an expression:

1. In the Expressions page, click the Add Expression button.

2. Enter the following information for the COUNT aggregate:

Page Element	***Value or Status***
Expression Type	*Number*
Aggregate Function	Selected
Length	*2*
Expression Text	*COUNT(A.ORDER_NBR)*

3. Click the OK button, and click the Use as Field link for the COUNT(A.ORDER_NBR) expression.
4. Save the query.

Adding the MAX Aggregate Function to an Expression

To add the MAX aggregate function to an expression:

1. In the Expressions page, click the Add Expression button.
2. Enter the following information for the MAX aggregate:

Page Element	***Value or Status***
Expression Type	*Number*
Aggregate Function	Selected
Length	*9*
Decimal	*2*
Expression Text	*MAX(A.PRICE)*

3. Click the OK button, and click the Use as Field link for the MAX(A.PRICE) expression.
4. Save the query.

Adding the MIN Aggregate Function to an Expression

To add the MIN aggregate function to an expression:

1. In the Expressions page, click the Add Expression button.

2. Enter the following information for the MIN aggregate:

Page Element	***Value or Status***
Expression Type	*Number*
Aggregate Function	Selected
Length	*9*
Decimal	*2*
Expression Text	*MIN(A.PRICE)*

3. Click the OK button, and click the Use as Field link for the MIN(A.PRICE) expression.
4. Save the query.

Adding the SUM Aggregate Function to an Expression

To add the SUM aggregate function to an expression:

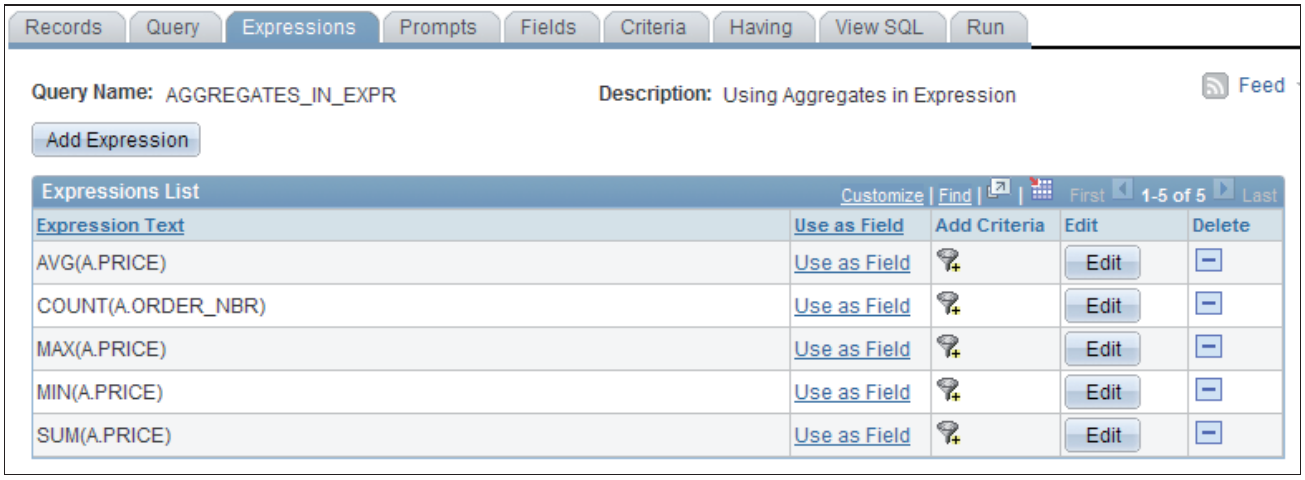
1. In the Expressions page, click the Add Expression button.
2. Enter the following information for the SUM aggregate:

Page Element	***Value or Status***
Expression Type	*Number*
Aggregate Function	Selected
Length	*9*
Decimal	*2*
Expression Text	*SUM(A.PRICE)*

3. Click the OK button, and click the Use as Field link for the SUM(A.PRICE) expression.
4. Save the query, and compare the query to the results.

Results

This example shows the Expressions page of the AGGREGATES_IN_EXPR query with multiple aggregates added:



Editing Fields and Previewing Results

To edit the fields and preview the results:

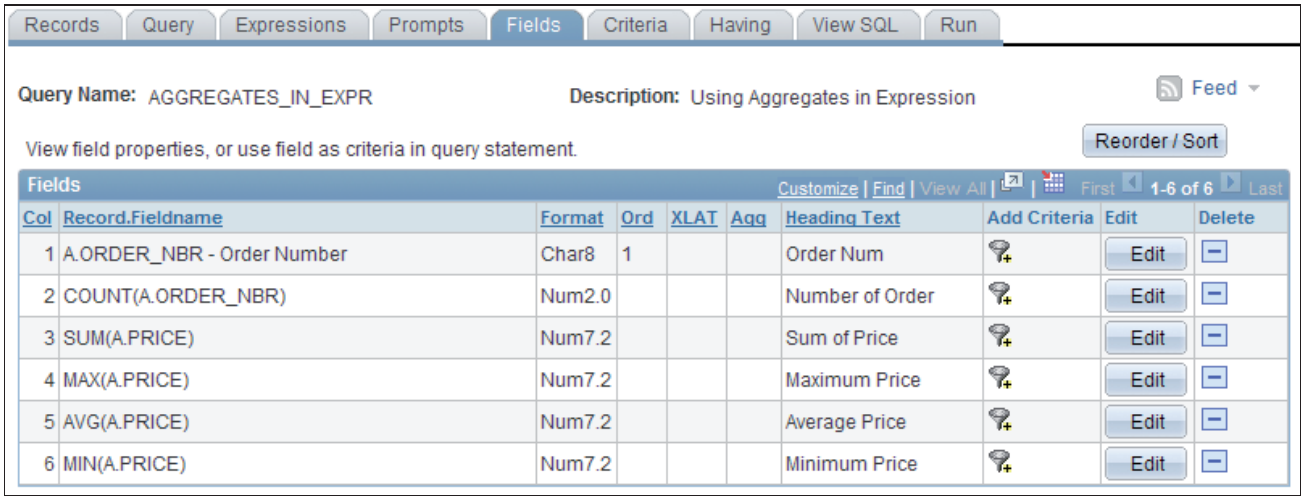
1. In the Fields page, editing the fields using the following information:

Field	***Value or Status***
A.ORDER_NBR	Order by 1 Column 1
AVG(A.PRICE)	*Average Price* Column 5
COUNT(A.ORDER_NBR)	*Number of Orders* Column 2
MAX(A.PRICE)	*Maximum Price* Column 4
SUM(A.PRICE)	*Sum of Price* Column 3
MIN(A.PRICE)	*Minimum Price* Column 6

2. Click the OK button, and save the query.
3. Compare the query output to the results.

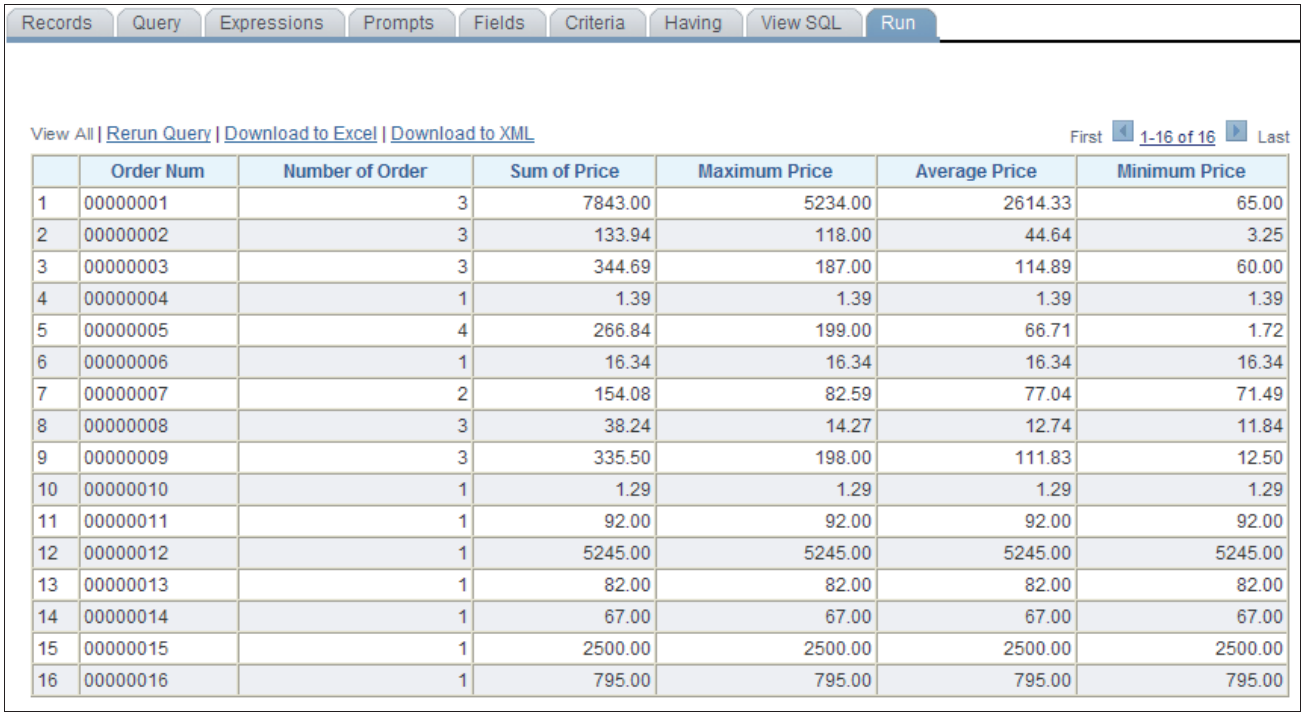
Results

This example shows the Fields page of the AGGREGATES_IN_EXPR query with multiple aggregate expressions used as fields:



Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete
1	A.ORDER_NBR - Order Number	Char8	1			Order Num		Edit	
2	COUNT(A.ORDER_NBR)	Num2.0				Number of Order		Edit	
3	SUM(A.PRICE)	Num7.2				Sum of Price		Edit	
4	MAX(A.PRICE)	Num7.2				Maximum Price		Edit	
5	AVG(A.PRICE)	Num7.2				Average Price		Edit	
6	MIN(A.PRICE)	Num7.2				Minimum Price		Edit	

This example shows the results of the AGGREGATES query, which displays multiple aggregate expressions as columns:



	Order Num	Number of Order	Sum of Price	Maximum Price	Average Price	Minimum Price
1	00000001	3	7843.00	5234.00	2614.33	65.00
2	00000002	3	133.94	118.00	44.64	3.25
3	00000003	3	344.69	187.00	114.89	60.00
4	00000004	1	1.39	1.39	1.39	1.39
5	00000005	4	266.84	199.00	66.71	1.72
6	00000006	1	16.34	16.34	16.34	16.34
7	00000007	2	154.08	82.59	77.04	71.49
8	00000008	3	38.24	14.27	12.74	11.84
9	00000009	3	335.50	198.00	111.83	12.50
10	00000010	1	1.29	1.29	1.29	1.29
11	00000011	1	92.00	92.00	92.00	92.00
12	00000012	1	5245.00	5245.00	5245.00	5245.00
13	00000013	1	82.00	82.00	82.00	82.00
14	00000014	1	67.00	67.00	67.00	67.00
15	00000015	1	2500.00	2500.00	2500.00	2500.00
16	00000016	1	795.00	795.00	795.00	795.00

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- You manage expressions using the Expressions and the Edit Expression Properties pages.
- You use expressions for mathematical calculations that are not available in the list of aggregate functions.
- You use expressions in criteria to narrow the query results.
- You use literals in expressions to insert spaces or other characters in the field values when you run the query.
- You use links in expressions to navigate to other websites, launch email, and launch documents.
- You use prompts in expressions when you want users to enter values at runtime.
- You include aggregates in expressions.

Slide 149

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Using aggregates in expressions Using expressions in criteria Using prompts in expressions	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Defining Selection Criteria"

Lesson 11

Drilling URLs in PeopleSoft Query

Objectives

By the end of this lesson, you will be able to:

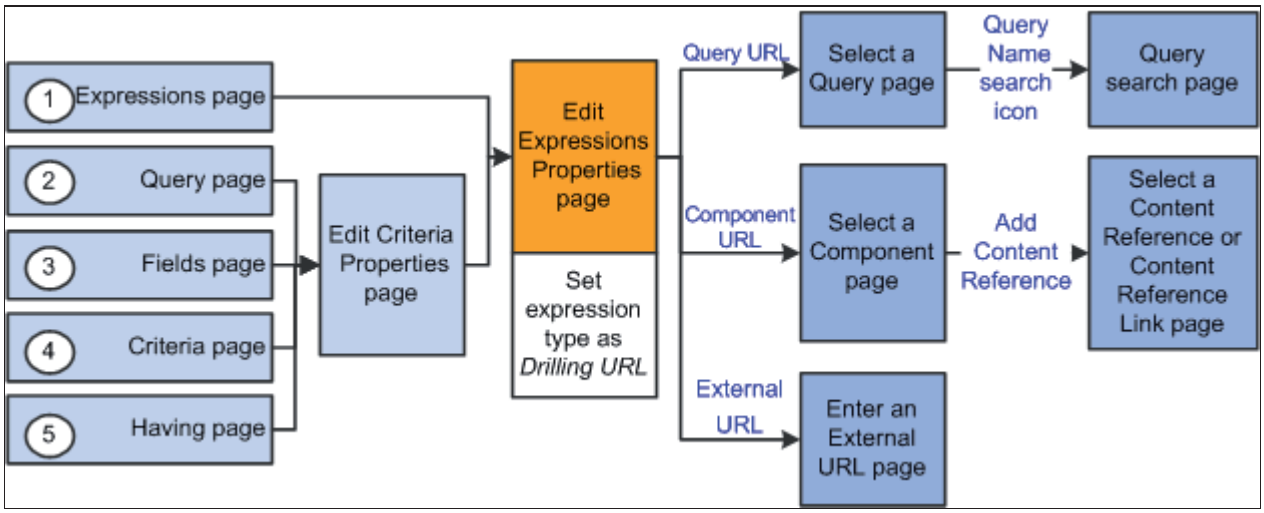
- Describe drilling URLs in PeopleSoft Query.
- View and edit expression properties.
- Run queries that have drilling URLs defined.
- Schedule queries that have drilling URLs defined.

Slide 151

Describing Drilling URLs in PeopleSoft Query

Understanding Drilling URLs

This diagram shows the navigation paths to access pages used to define drilling URLs in PeopleSoft Query:



Slide 152

Student Notes

Understanding Drilling URLs

Drilling URLs are the URLs that you define by selecting the menu, component, page, portal object, or URL of choice.

When you build a query using Query Manager, you can define drilling URLs that are associated with this query. These settings are saved into the database, along with prompt, criteria, and so on, as part of the metadata for this query. When you execute this query through Query Manager or Query Viewer, the query results page shows results as links, which you can click to be redirected to a different page in a new browser window.

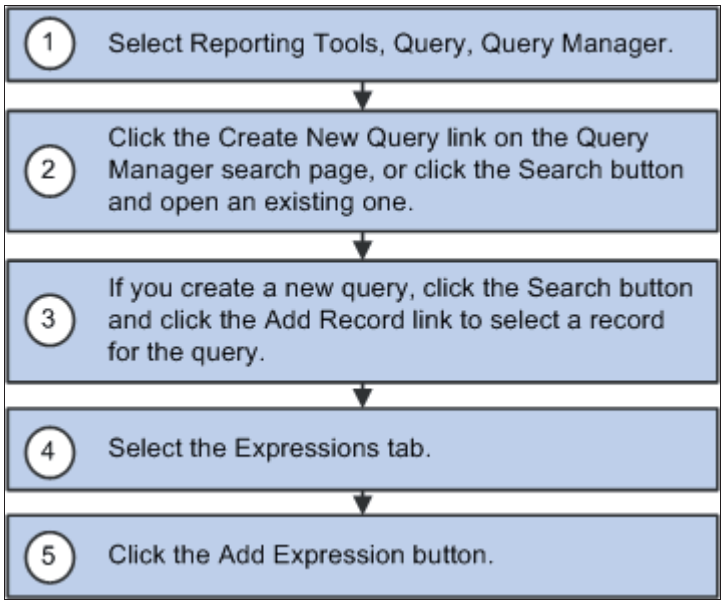
Depending on how drilling URLs are defined, the new browser is either a PeopleSoft Pure Internet Architecture page, another query result page, or an external page.

Viewing and Editing Expression Properties

Viewing and Editing Expression Properties

Drilling URLs are a special type of expression that you can define using the Edit Expression Properties page in Query Manager.

This diagram shows steps of how to access the Edit Expression Properties page:



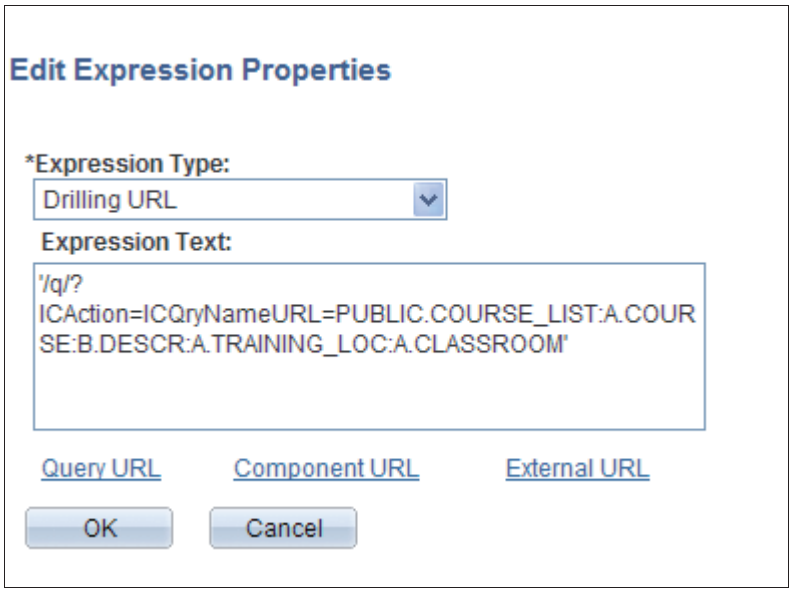
Slide 153

Student Notes

Page Used to View and Edit Expression Properties

Use this page to view and edit expression properties:

Page Name	*Navigation*
Edit Expression Properties	1. Select Reporting Tools, Query, Query Manager. 2. Create a new query or open an existing one. 3. Select the Expressions tab, and click the Add Expression button.



Elements of the Edit Expression Properties Page

The elements of the Edit Expression Properties page are:

Expression Type	To define drilling URLs, you must select the *Drilling URL* option from the Expression Type drop-down list box.
Expression Text	Click the Query URL, Component URL, or External URL links to allow the appropriate system building URLs. Alternately, type the URL in the Expression Text box.
	Note. If you type the URL directly into the Expression Text box, the system does not validate against a value for the correct format.
Query URL	Click to access the Query URL definition widget, where you can select a query to build URLs in a query URL format.
Component URL	Click to access the Component URL definition widget, where you can select a component to build URLs in a component URL format.
External URL	Click to access the External URL definition widget, where you can enter external URL to build URLs in an external URL format.

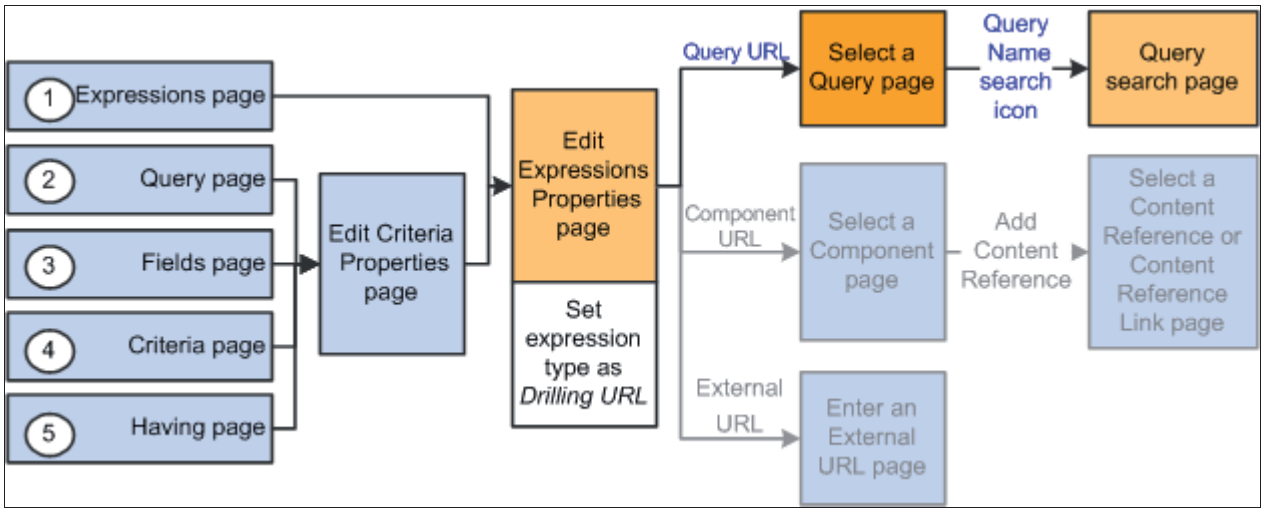
Note. Drilling URLs are saved into database as an expression, so you have the option of accessing the Expressions page and adding the defined drilling URLs as fields. However, since drilling URLs are a special type of expression, you cannot add it as a criterion. You can add drilling URLs as query fields just like regular expressions. On a query result page, values in that column will be expanded to a fully qualified URL, which you can click to either run a query, access a PeopleSoft Pure Internet Architecture page, or go to an external URL.

Viewing and Editing Expression Properties (continued)

Defining Query URL widgets

The Query URL widget enables you to use the Select a Query page to build drilling URLs in a query URL format.

This diagram shows steps used to define query URL widgets:



Slide 154

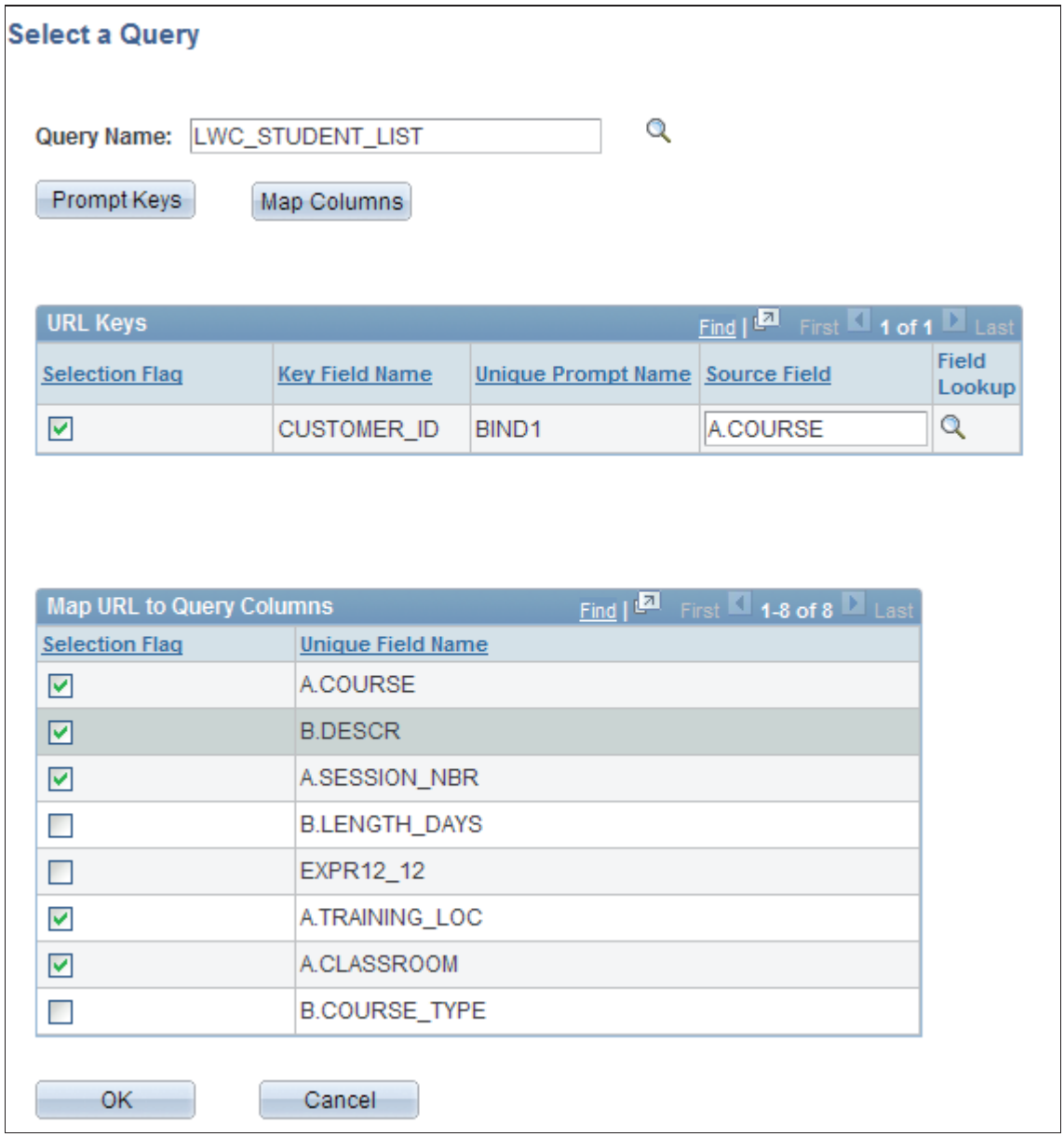
Student Notes

Pages Used to Define Query URL Widgets

Use these pages to define query URL widgets:

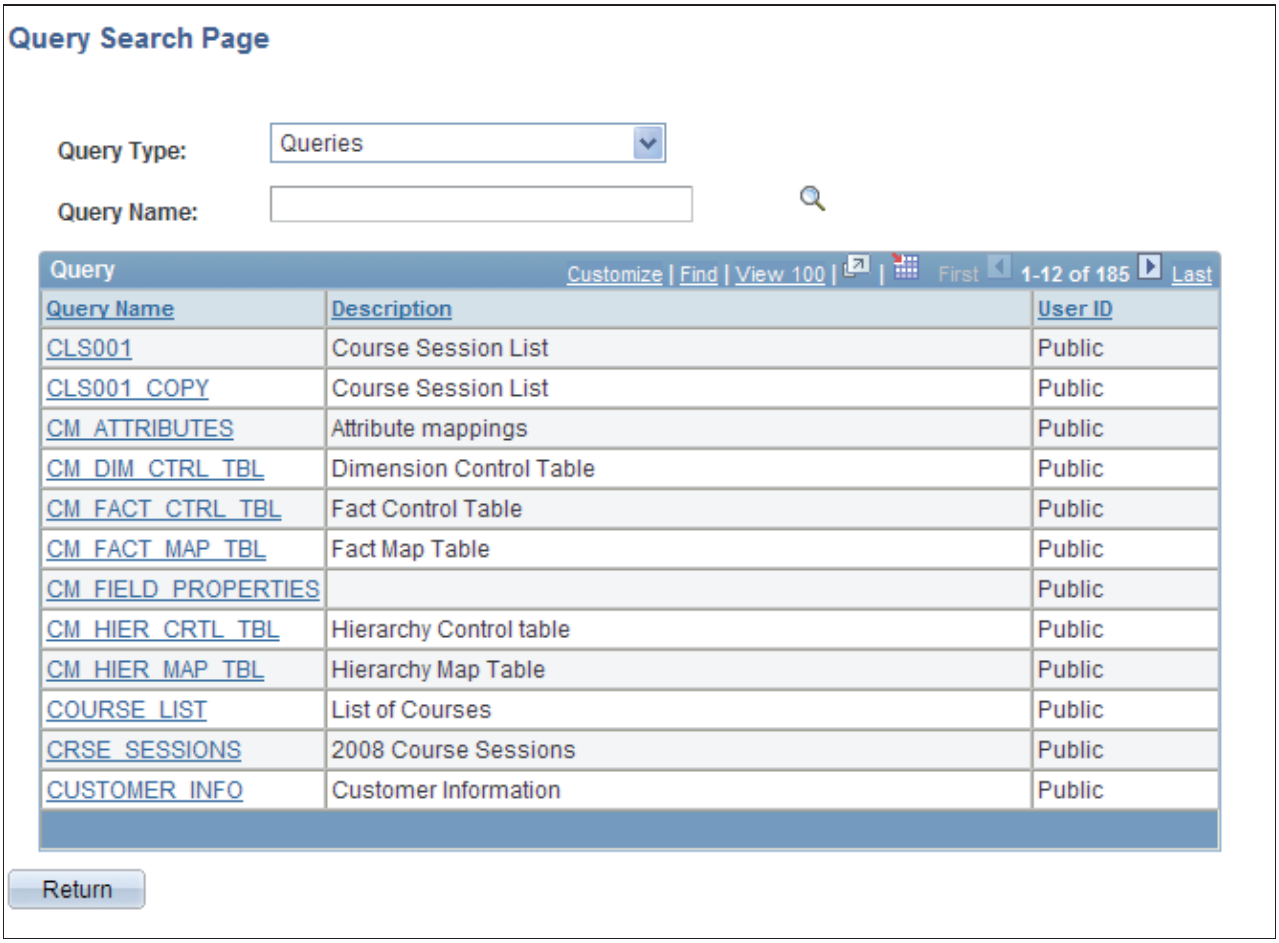
- Use this page to select a query to build drilling URLs in a query URL format:

Page Name	*Navigation*
Select a Query	1. In Query Manager, select the Expressions tab, and click the Add Expression button. 2. Select the Drilling URL option from the Expression Type drop-down list box, and click the Query URL link.



- Use this page to search for a prompt criteria of the query to build URLs:

Page Name	*Navigation*
Query Search Page	From the Select a Query page, click the Prompt Key icon next to the Query Name field.



Elements of the Select a Query Page

The elements of the Select a Query page are:

Query Name Type a query name in the text box or click the query lookup icon to search for an existing query.

Prompt Keys

Click to access the Query Search Page where you can search for the prompt criteria of the query that you entered in the Query Name field.

If the entered query has prompt criteria, a list of those fields appears in the URL Keys section. If the entered query has no prompt criteria, a message appears saying *Query does not contain any prompt key(s).*

Note. This step is optional. If you do not map any prompt keys to the source query column field, then when you click a drilling URL link in query result column page, you will be directed to the Prompts page, where you can enter prompt key values before retrieving query result.

Map Columns

Click to display the Map URL to Query Columns section.

Note. This step is required for the drilling URLs to be available in a query results page.

You can define multiple drilling URLs in a query, but each query result column can only bind with one URL. Therefore, when you click the Map Column button, only those fields that have no bond with any URL appear in the Map URL to Query Columns section. If all query columns have bonds with other tracking URLs, then a message appears saying *All columns have already been mapped to other drilling URLs.*

URL Keys

The URL Keys section is available only after you define the Query field and click the Prompt Keys button.

In this section, you define value mapping between Drilling URL destination query prompt fields and Drilling URL source query result column fields. To map these fields, select the key field by selecting the check box adjacent to a prompt key field, and then click the lookup icon to select the source query column field to map to it.

Note. This step is optional. If you do not map any prompt keys to a source query column field, then when you click a drilling URL link on the query result column page, you will be directed to the Prompt page, where you can enter prompt key values before you retrieve query results.

Map URL to Query Columns

The Map URL to Query Columns section is available only after you define the component values and click the Map Columns button.

Note. This step is required for the drilling URLs to be available in the query results page.

Use the Map Columns button for binding source query result column fields with Drilling URL. You must map URL to query result columns in order for the drilling URLs to be available as links in column fields of the source query results. You can define column mapping by selecting the appropriate check box in the Map URL to Query Columns section.

Notes of Query Drilling URLs

Note that, a query drilling URL:

- Always begins with '/q/.

- Is saved as part of expression metadata for the query.
- Contains two parts: query URL format and query result columns binding with the Drilling URL.

Steps Used to Select a Query to Build Drilling URL

To select a query to build drilling URLs:

1. Select Reporting Tools, Query, Query Manager.
2. Create a new query or search for an existing one.
3. Select the Expressions tab, and click the Add Expression button.
4. Select the *Drilling URL* option from the Expression Type list.
5. Click the Query URL link to define drilling URL for the source query that you just created.
6. On the Select a Query page, type a query name in the Query Name text box.

   Alternatively, click the lookup icon to search for one from existing queries.
7. Optionally, define value mapping in the URL Keys section.
   a. Click the Prompt Keys button to bind the prompt keys of the destination query with the columns of the source query.

   Note. If the destination query has prompt criteria, the URL Keys section appears with a list of key fields.

   b. Define value mapping in the URL Keys section by selecting the check box adjacent to a prompt key field, and then clicking the lookup icon to select a source query column field to map to it.
8. Optionally, define the column mapping in the Map URL to Query Columns section:
   a. Click the Map Columns button to map the drilling URL to a source query selected column.
   b. Select appropriate values from the Map URL to Query Columns section.
9. Click the OK button.

   The Edit Expression Properties reappears with query drilling URLs in the Expression Text field.
10. Save your query and view the results.

    The query results page displays results as links. When you click these links, the destination query is run using prompt key values that are defined using the source query.

Explanation of the Query Drilling URL

After you build the query drilling URL, the Edit Expression Properties reappears with the query drilling URLs in the Expression Text field. For example:

```
'/q/?ICAction=ICQryNameURL=PUBLIC.DESTINATION&BIND1=A.DEPTID:A.DEPTID'
```

In this query drilling URL, the standard query URL format is: `/q/?ICAction=ICQryNameURL=PUBLIC.DESTINATION&BIND1=A.DEPTID` and the binding column is `A.DEPTID`.

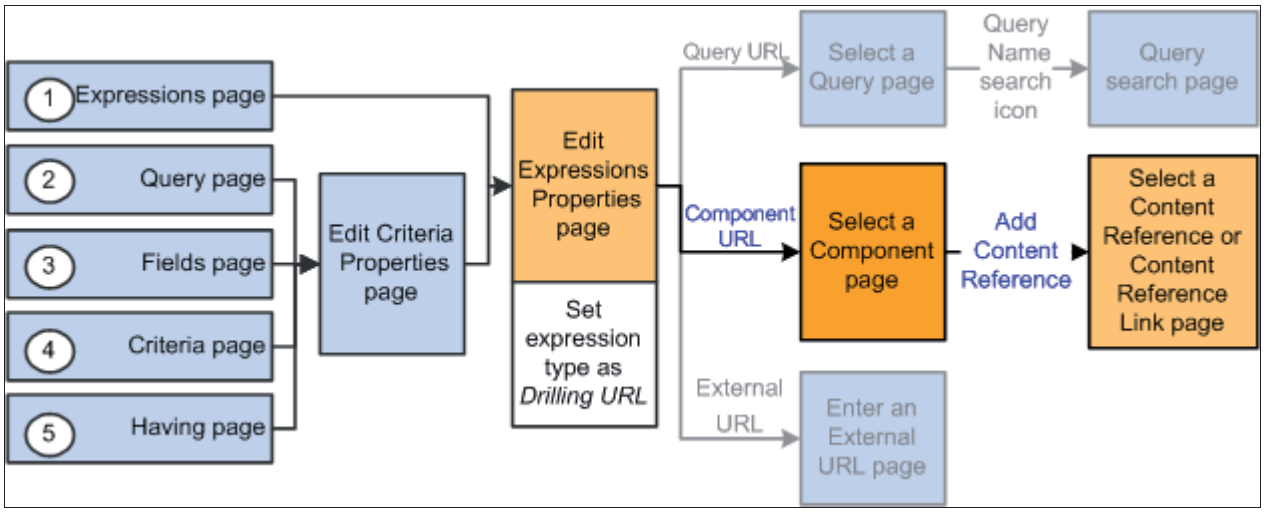
Thus, this query drilling URL is binding with column A.DEPTID.

Viewing and Editing Expression Properties (continued)

Defining Component URL Widgets

The Component URL widget enables you to use the Select a Component page to build drilling URLs in component URL format.

This diagram shows steps used to define component URL widgets:



Slide 155

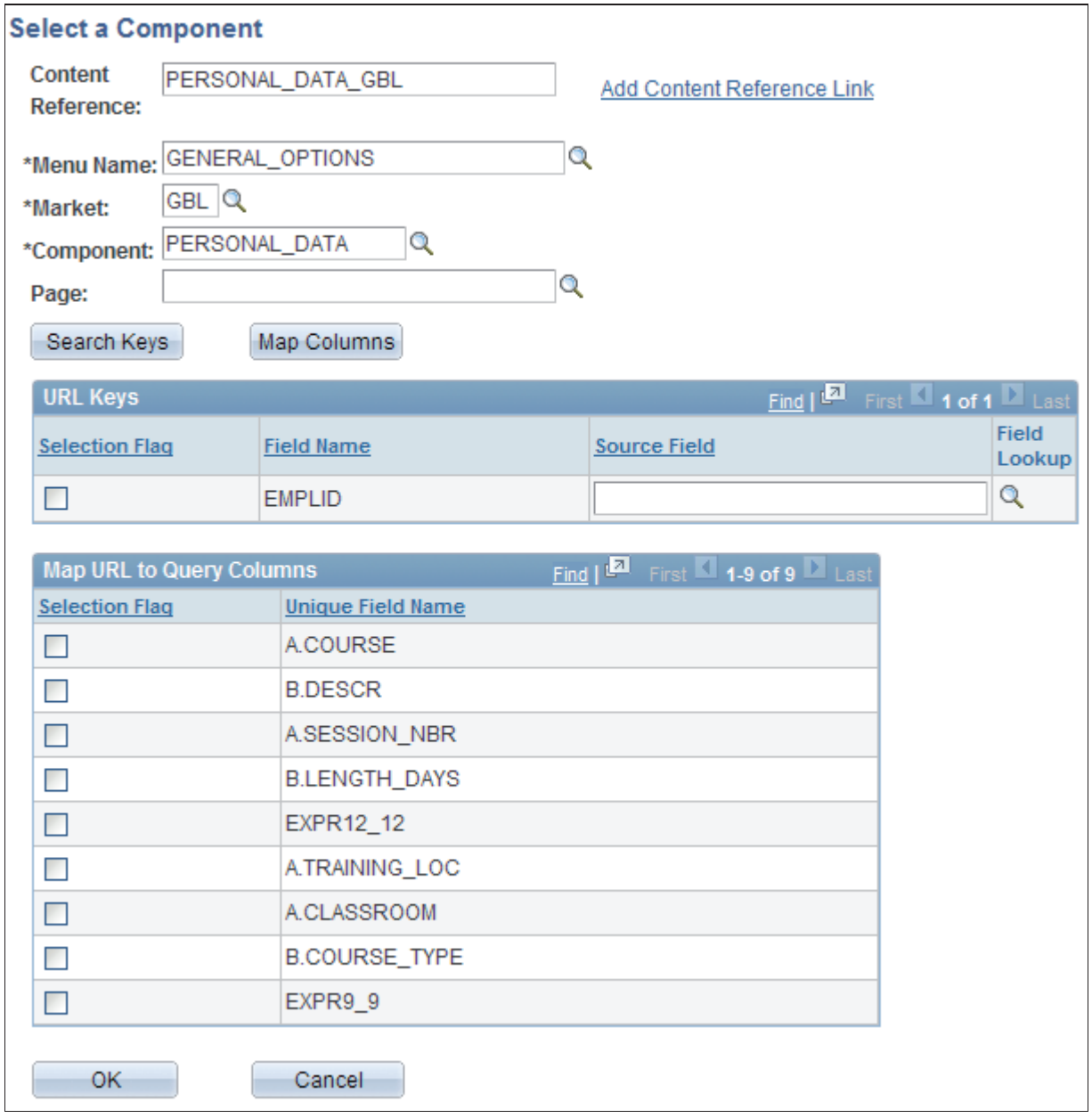
Student Notes

Pages Used to Defining Component URL Widgets

Use these pages to define component URL widgets:

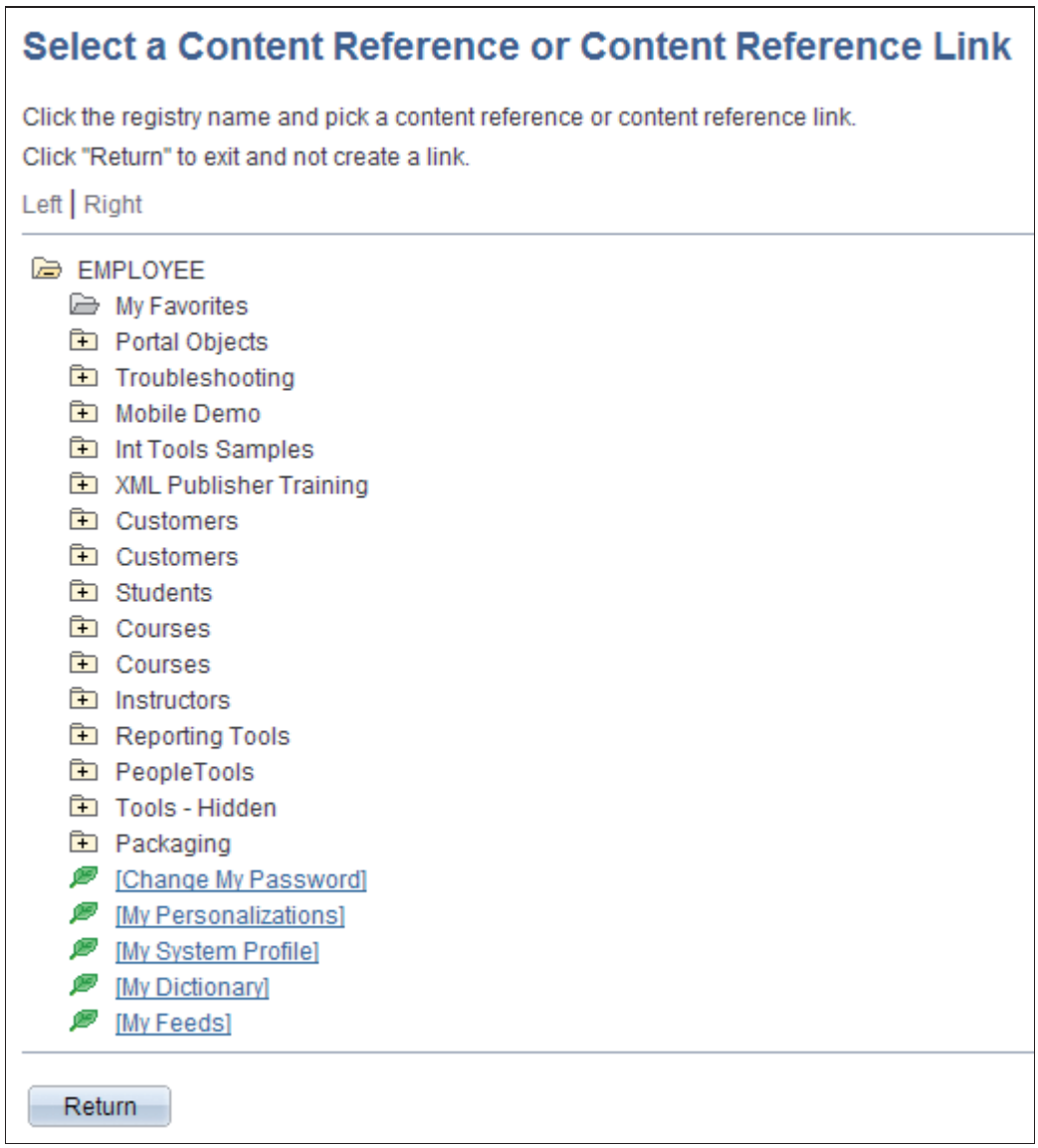
- Use this page to select a component to build drilling URLs in a component URL format:

Page Name	*Navigation*
Select a Component	1. In Query Manager, select the Expressions tab, and click the Add Expression button. 2. Select the *Drilling URL* option from the Expression Type list, and click the Component URL link.



- Use this page to select a content reference or content reference link:

Page Name	*Navigation*
Select a Content Reference or Content Reference Link	From the Select a Component page, click the Add Content Reference Link link.



Elements of the Select a Component Page

The elements of the Select a Component page are:

Content Reference Enter the name of a content reference.

Add Content Reference Link

Click the Add Content Reference Link link to access the Select a Content Reference or Content Reference Link page, where you can select a content reference or content reference link from an existing list.

Note. The values of the Menu Name, Market, and Component fields are populated if you select a content reference or content reference link using the Add Content Reference Link link.

Menu Name

Enter a menu name or click the lookup icon and select one menu name from an existing list.

Otherwise, the menu name value is populated if you select a content reference or content reference link using the Add Content Reference Link link.

Market

Enter a market code or click the lookup icon and select one market code from an existing list.

Otherwise, the market code is populated automatically if you select a content reference or content reference link using the Add Content Reference Link link.

Component

Enter a component name or click the lookup icon and select one component name from an existing list.

Otherwise, the component name value is populated if you select a content reference or content reference link using the Add Content Reference Link link.

Search Keys

Click to select mapping between component search keys and source query result columns.

If the entered component has a search key, a list of those search keys appears in the URL Keys section. If the entered component has no search key, a message appears saying *Component does not contain any search keys.*

Note. This step is optional. If you do not map any search keys to the source component, then when you click a drilling URL link in the query result column page, you will be directed to a component search page, where you can select search key values.

Map Columns

Click to display the Map URL to Query Columns section.

Note. This step is required for the drilling URLs to be available in query results page.
You can define multiple drilling URLs in a query, but each query result column can only bind with one URL. When you click the Map Columns button, only those fields that have no bond with any URL appear in the Map URL to Query Columns section. If all query columns have bonds with other tracking URLs, then a message appears saying *All columns have already been mapped to other drilling URLs.*

URL Keys

The URL Keys section is available only after you define the component and click the Search Keys button.

In this URL Keys section, you define value mapping between Drilling URL destination component search keys and Drilling URL source query result column fields.

To map these fields, select the key field by selecting its check box, and then click the lookup icon to select source query column field to map to it.

Note. This step is optional. If you do not map any search keys to the source component, then when you click a drilling URL link in query result column page, you will be directed to a component search page, where you can enter search key values before you launch the component.

Map URL to Query Columns

The Map URL to Query Columns section is available only after you define the component values and click the Map Columns button.

Note. This step is required for the drilling URLs to be available in the query results page.
Use the Map Columns button for binding source query result column fields with a Drilling URL. You must map URLs to query result columns in order for the drilling URLs to be available as links in column fields of the source query results. You can define column mapping by selecting the appropriate check box in this Map URL to Query Columns section.

Notes of Component Drilling URLs

Note that, a component drilling URL:

- Always begins with `'/c/`.
- Is saved as part of expression metadata for the query.
- Contains two parts: component URL format and query result columns binding with the drilling URL.

Steps Used to Select a Component to Build Drilling URLs

To select a component to build drilling URLs:

1. Select Reporting Tools, Query, Query Manager.
2. Create a new query, or open an existing one.
3. Select the Expressions tab, and click the Add Expression button.
4. Select the *Drilling URL* option from the Expression Type drop-down list box.
5. Click the Component URL link to define a drilling URL for the source query that you just created.
6. Type a content reference in the text box.

   Alternatively, click the Add Content Reference Link link to select one from existing components.
7. If needed, define the values for the Menu Name, Market, and Component fields.

8. Optionally, define value mapping in the URL Keys section.

   a. Click the Search Keys button.

   Note. If the selected component has a search key, the URL Keys section appears with a list of search key fields.

   b. Define value mapping in the URL Keys section by selecting an appropriate search field and selecting a source query results column.

9. Optionally, define the map columns in the Map URL to Query Columns section.

   a. Click the Map Columns button to map the drilling URL to a source query selected column.

   b. Select appropriate values from the Map URL to Query Columns section.

10. Click the OK button.

    The Edit Expression Properties page reappears with a component drilling URL in the Expression Text field.

11. Save your query and preview the query results.

    The query results page displays results as links. When you click these links, destination query is run using the prompt key value that is defined using the source query.

Explanation of the Component Drilling URL

After you build the component drilling URLs, the Edit Expression Properties reappears with the query drilling URLs in the Expression Text field. For example:

```
'/c/QE_SAMPLE_APPS.QE_DEPT_TBL.GBL?Action=U&DEPTID=A.DEPTID&SETID=A.SETID
:A.SETID:
A.DEPTID'
```

In this component drilling URL, the standard query URL format is:

```
/c/QE_SAMPLE_APPS.QE_DEPT_TBL.GBL?Action=U&DEPTID=A.DEPTID&SETID=A.SETID
```

and the binding columns is `A.SETID:A.DEPTID`.

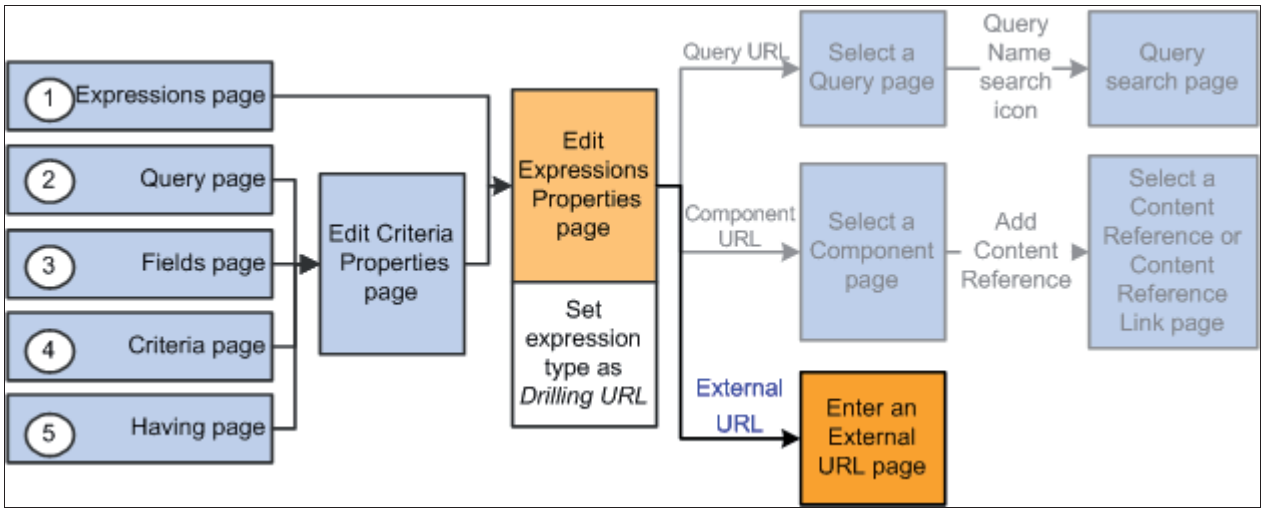
Thus, this drilling URL is binding with column `A.DEPTID` and `A.SETID`.

Viewing and Editing Expression Properties (continued)

Defining External URL Widgets

The External URL widget enables you to use the Enter an External URL page to build drilling URLs in an external URL format.

This diagram shows steps used to define external URL widgets:



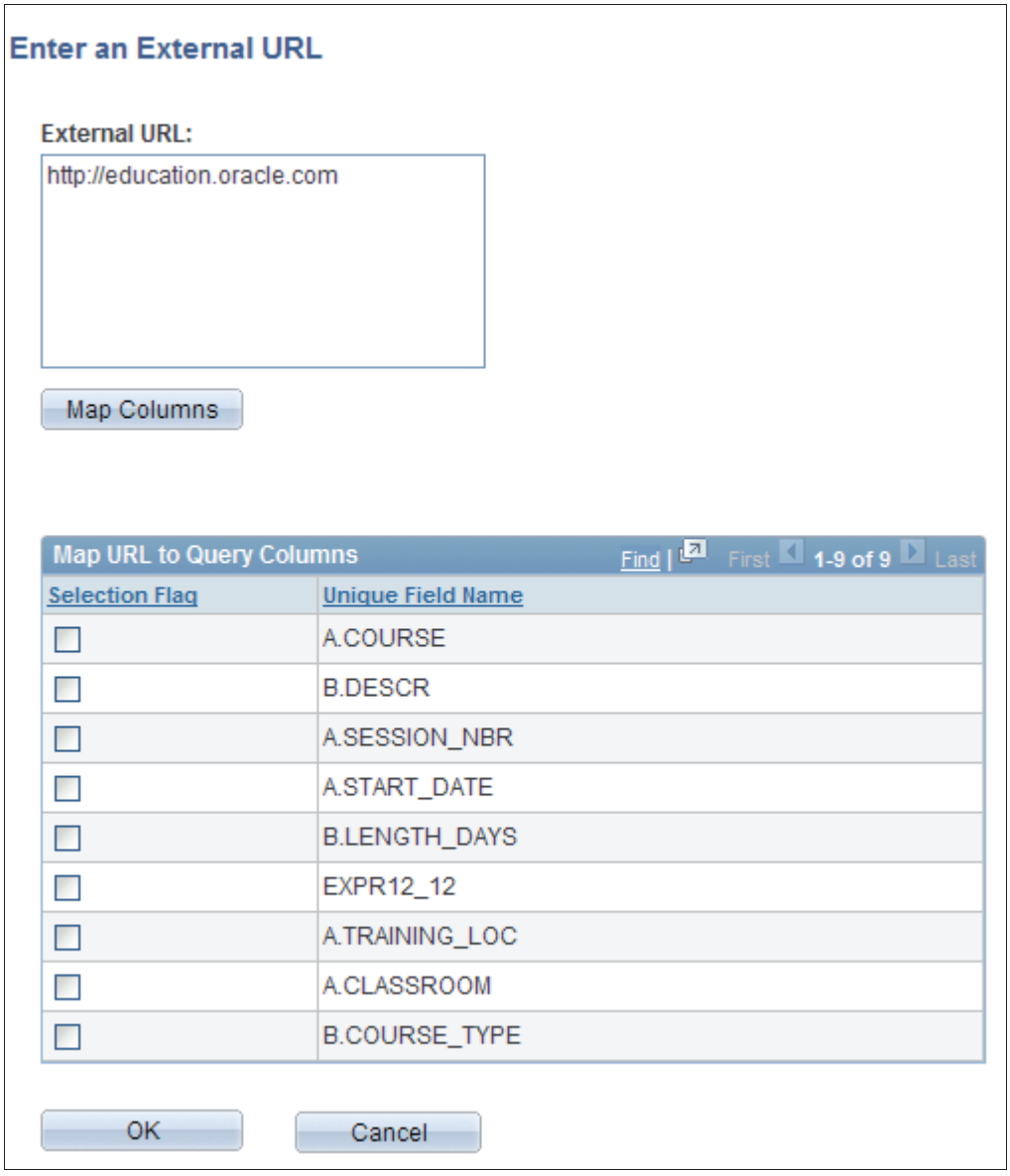
Slide 156

Student Notes

Page Used to Define External URL Widgets

Use this page to enter an external URL to build drilling URLs in external URL format:

Page Name	*Navigation*
Enter an External URL	1. In Query Manager, select the Expressions tab, and click the Add Expression button. 2. Select the *Drilling URL* option from the Expression Type list, and click the External URL link.



Note. Query URL and component URL have a certain format for key list values in the URL. However, external URL has no key mapping widget because external URL is free-format URL.

Elements of the Enter an External URL Page

The elements of the Enter an External URL page are:

External URL	Enter the value of the external URL. There is no format restriction.
Map Columns	After you enter URL value, click the Map Columns button to display the Map URL to Query Columns section.

Note. This step is required for the drilling URLs to be available in the query results page.
You can define multiple drilling URLs in a query, but each query result column can only bind with one URL. When you click the Map Column button, only those fields that have no bond with any URL appear in the Map URL to Query Columns section. If all query columns have bonds with other tracking URLs, then a message appears saying *All columns have already been mapped to other drilling URLs.*

Map URL to Query Columns	This section is available only after you define the component values and click the Map Columns button.

Note. This step is required for the drilling URLs to be available in query results page.
Use the Map Columns button for binding source query result column fields with a Drilling URL. You must map URL to query result columns in order for the drilling URLs to be available as links in column fields of source query results. You can define column mapping by selecting an appropriate check box in this Map URL to Query Columns section.

Steps Used to Enter External URL to Build Drilling URLs

To enter an external URL to build drilling URLs:

1. Select Reporting Tools, Query, Query Manager.
2. Create a new query or open an existing one.
3. Select the Expressions tab, and click the Add Expression button.
4. Select the *Drilling URL* option from the Expression Type drop-down list box.
5. Click the External URL link to define drilling URL for the source query that you just created.
6. Type a valid URL in the External URL edit box.
7. Optionally, define the column mapping in the Map URL to Query Columns section.
   a. Click the Map Columns button to map the drilling URL to source query selected column.
   b. Select appropriate values from the Map URL to Query Columns section.
8. Click the OK button.

   The Edit Expression Properties reappears with query drilling URLs in the Expression Text field.
9. Save your query and preview the query results.

   The query results page displays results as links. When you click these links, destination query is run using prompt key value that is defined using the source query.

Notes of External Drilling URL

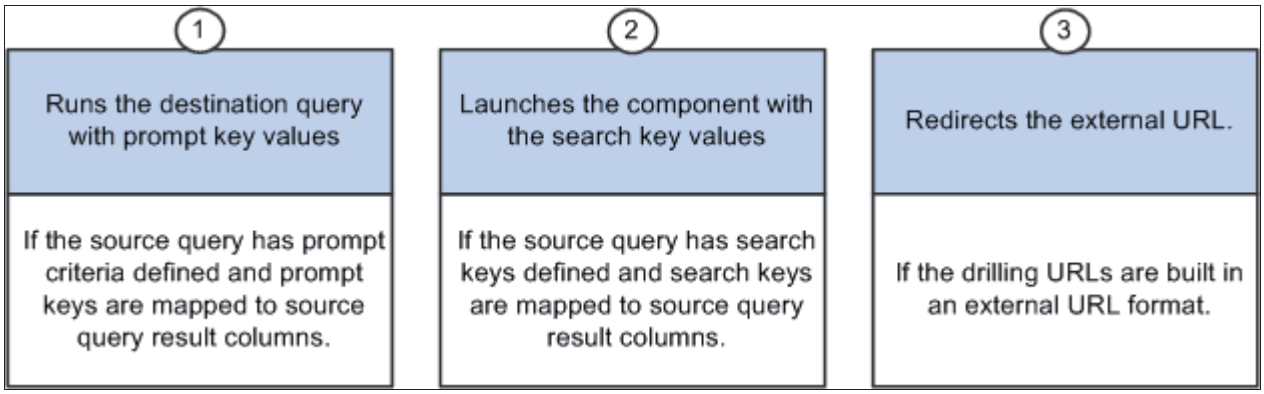
An external drilling URL always begins with `'/e/`.

This is an example of external drilling URL:
`'/e/?url=[http://education.oracle.com]:A.COURSE'`.

Running Queries that Have Drilling URLs Defined

Query Results with Drilling URLs

When you access the Run page of a query that has a Drilling URL defined, its query results are shown as links. Clicking these result links either:



Slide 157

Student Notes

Example of the Run page with Query Results as Links

This is an example of the Run page with query results as links:

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

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

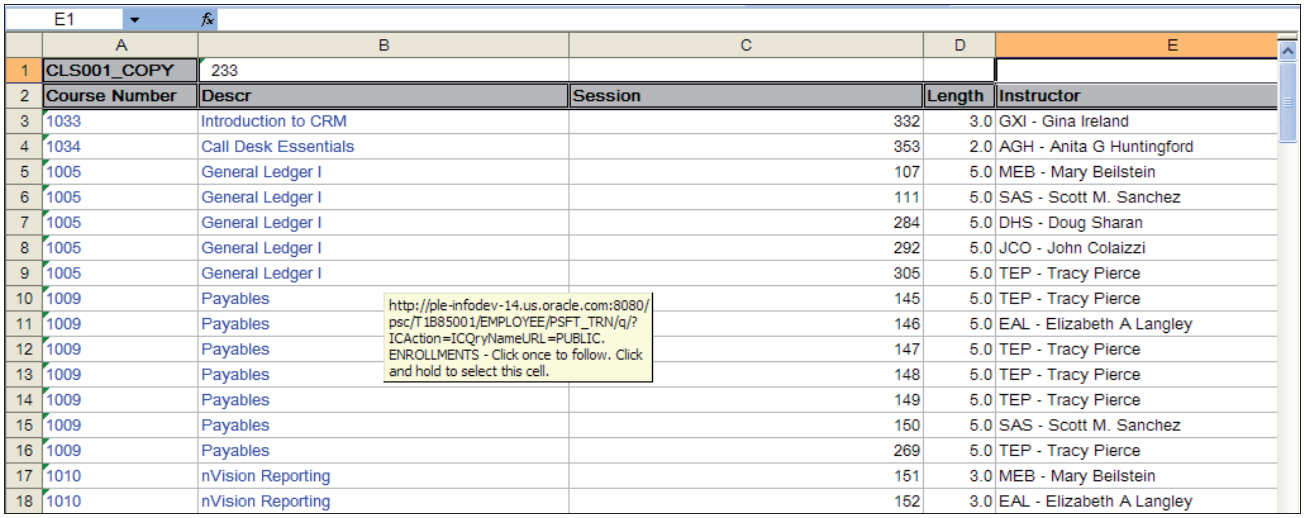
First 1-100 of 233 Last

	Course Number	Descr	Session	Length	Instructor	Trn Loc	Room	Type
1	1033	Introduction to CRM	332	3.0	GXI - Gina Ireland	STH	X	CRM
2	1034	Call Desk Essentials	353	2.0	AGH - Anita G Huntingford	BOS	X	CRM
3	1005	General Ledger I	107	5.0	MEB - Mary Beilstein	TEA	C	Financials
4	1005	General Ledger I	111	5.0	SAS - Scott M. Sanchez	CORP	E	Financials
5	1005	General Ledger I	284	5.0	DHS - Doug Sharan	ATL	A	Financials
6	1005	General Ledger I	292	5.0	JCO - John Colaizzi	TEA	A	Financials
7	1005	General Ledger I	305	5.0	TEP - Tracy Pierce	TEA	A	Financials
8	1009	Payables	145	5.0	TEP - Tracy Pierce	ONSTE	O	Financials
9	1009	Payables	146	5.0	EAL - Elizabeth A Langley	WC	D	Financials
10	1009	Payables	147	5.0	TEP - Tracy Pierce	ONSTE	O	Financials

Note. On the Run page, when you click the Download to Excel link to download query results to a Microsoft Excel spreadsheet, an Excel spreadsheet is launched with links to respective cells. When you click the Excel link from Query Manager or the Query Viewer search results page, you get similar query results.

Example of Query Results with Drilling URLs in Microsoft Excel

This is an example of query results in Microsoft Excel when you click the Excel link in Query Manager or Query Viewer. Notice that the URL appears when you hover the mouse over a cell:

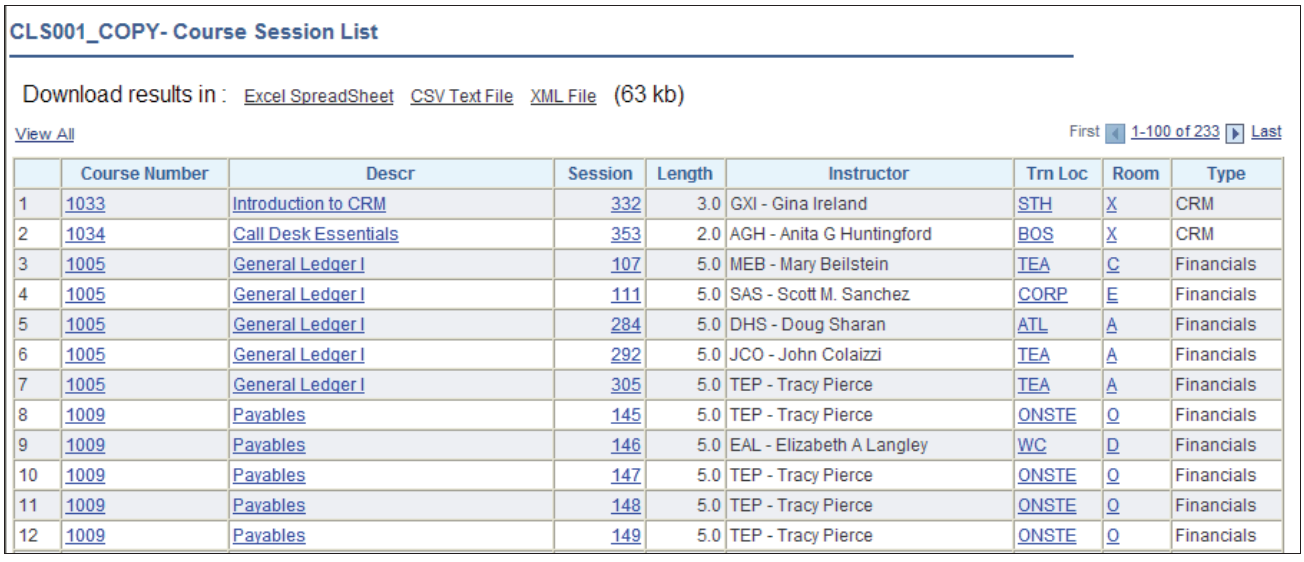


	A	B	C	D	E
1	CLS001_COPY	233			
2	Course Number	Descr	Session	Length	Instructor
3	1033	Introduction to CRM	332	3.0	GXI - Gina Ireland
4	1034	Call Desk Essentials	353	2.0	AGH - Anita G Huntingford
5	1005	General Ledger I	107	5.0	MEB - Mary Beilstein
6	1005	General Ledger I	111	5.0	SAS - Scott M. Sanchez
7	1005	General Ledger I	284	5.0	DHS - Doug Sharan
8	1005	General Ledger I	292	5.0	JCO - John Colaizzi
9	1005	General Ledger I	305	5.0	TEP - Tracy Pierce
10	1009	Payables	145	5.0	TEP - Tracy Pierce
11	1009	Payables	146	5.0	EAL - Elizabeth A Langley
12	1009	Payables	147	5.0	TEP - Tracy Pierce
13	1009	Payables	148	5.0	TEP - Tracy Pierce
14	1009	Payables	149	5.0	TEP - Tracy Pierce
15	1009	Payables	150	5.0	SAS - Scott M. Sanchez
16	1009	Payables	269	5.0	TEP - Tracy Pierce
17	1010	nVision Reporting	151	3.0	MEB - Mary Beilstein
18	1010	nVision Reporting	152	3.0	EAL - Elizabeth A Langley

Example of Query Results with Drilling URLs in HTML

You can also view the query results of a query that has drilling URL defined using the HTML links in Query Manager or Query Viewer.

This is an example of query results as links when you click the HTML links in Query Manager or Query Viewer:



CLS001_COPY- Course Session List

Download results in : Excel SpreadSheet CSV Text File XML File (63 kb)

View All First 1-100 of 233 Last

	Course Number	Descr	Session	Length	Instructor	Trn Loc	Room	Type
1	1033	Introduction to CRM	332	3.0	GXI - Gina Ireland	STH	X	CRM
2	1034	Call Desk Essentials	353	2.0	AGH - Anita G Huntingford	BOS	X	CRM
3	1005	General Ledger I	107	5.0	MEB - Mary Beilstein	TEA	C	Financials
4	1005	General Ledger I	111	5.0	SAS - Scott M. Sanchez	CORP	E	Financials
5	1005	General Ledger I	284	5.0	DHS - Doug Sharan	ATL	A	Financials
6	1005	General Ledger I	292	5.0	JCO - John Colaizzi	TEA	A	Financials
7	1005	General Ledger I	305	5.0	TEP - Tracy Pierce	TEA	A	Financials
8	1009	Payables	145	5.0	TEP - Tracy Pierce	ONSTE	O	Financials
9	1009	Payables	146	5.0	EAL - Elizabeth A Langley	WC	D	Financials
10	1009	Payables	147	5.0	TEP - Tracy Pierce	ONSTE	O	Financials
11	1009	Payables	148	5.0	TEP - Tracy Pierce	ONSTE	O	Financials
12	1009	Payables	149	5.0	TEP - Tracy Pierce	ONSTE	O	Financials

In this query results page:

- You can click the Excel Spreadsheet link to download query results to Microsoft Excel.

Note. The Excel Spreadsheet link in this query result page has the same usage as the Excel link in Query Manager or the Query Viewer search results page.

- You can click the CSV Text File link to download query results into a CSV format file.

Note. Unlike HTML and Excel, no links exist in the CSV format file because CSV file is opened using a text editor such as Notepad and links do not apply to text editors. However, if you select drilling URL as query selected field, then the drilling URL appears in a fully expanded version.

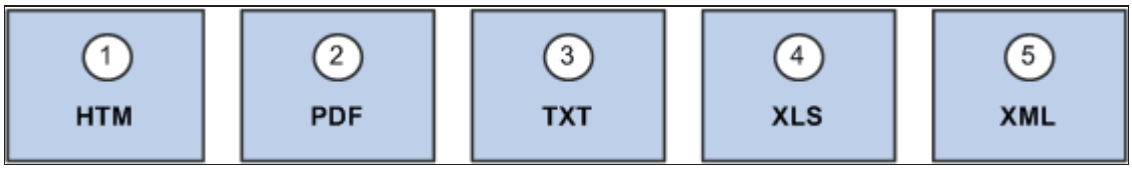
This is an example of CSV file content:

```
"QEDM1","10900","10900","http://buffy-
lap1.peoplesoft.com:8080/psp/ps_2/EMPLOYEE
/QE_LOCAL/c/QE_SAMPLE_APPS.QE_DEPT_TBL.GBL?Action=U&DEPTID=10900&SETID=QE
DM1
```

Scheduling Queries that Have Drilling URLs Defined

Schedule Query Page

Five different output types of the Schedule Query page are:

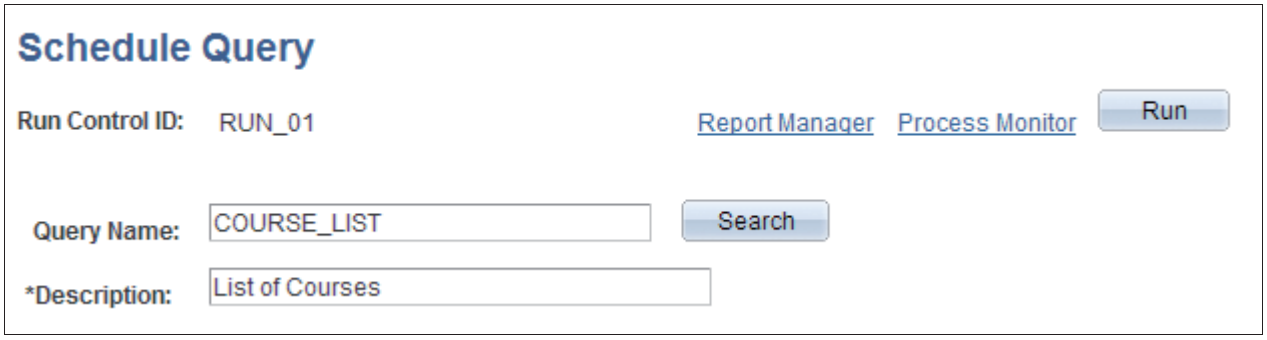


Slide 158

Page Used to Schedule Queries

Use this page to submits a process request and brings up the Process Request page, which enables you to specify such variables as where to run the process and in what format to generate the output based on a run control ID:

Page Name	*Navigation*
Schedule Query	Reporting Tools, Query, Schedule Query



Scheduling Queries

After you click the Run button on the Schedule Query page, the Process Scheduler Request page appears. This page has five different output types: *HTM, PDF, TXT, XLS,* and *XML.*

Result files from a successful schedule query run contain links for respective columns using *HTM,PDF,* and *XLS* formats.

If drilling URL is defined in the query, then no links exist in the result file for *TXT* and *XML* formats. However, if you select drilling URLs as a query selected field, drilling URLs are shown in a fully expanded version.

This is an example of TXT file content:

```
"QEDM1","10900","10900","http://buffy-
lap1.peoplesoft.com:8080/psp/ps_2/EMPLOYEE
/QE_LOCAL/c/QE_SAMPLE_APPS.QE_DEPT_TBL.GBL?Action=U&DEPTID=10900&SETID=QE
DM1
```

Note. Theoretically, one query can have multiple drilling URLs defined; however, the number of URLs for each query should be limited to enhance performance. For queries that return a large number of results, having multiple URLs defined would slow query execution time.

Activity 21: Building Drilling URLs in PeopleSoft Query

In this activity, you will review the activity overview and:

1. Define Query URL widgets for an existing query.
2. View drilling URL results.

Slide 159

Activity Overview

Open the existing query CLS001, and save it as CLS001_DRILLING_URL.

Define query drilling URLs using the REVENUE query with the COURSE field as a prompt key field and the A.COURSE field as a source field.

Map the drilling URL to a source query's selected columns—A.COURSE, A.SESSION_NBR, and A.TRAINING_LOC.

When you drill the query results, use course *1002* with the enrollment dates between *07/01/2009* and *12/31/2009*.

Note. Use the *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

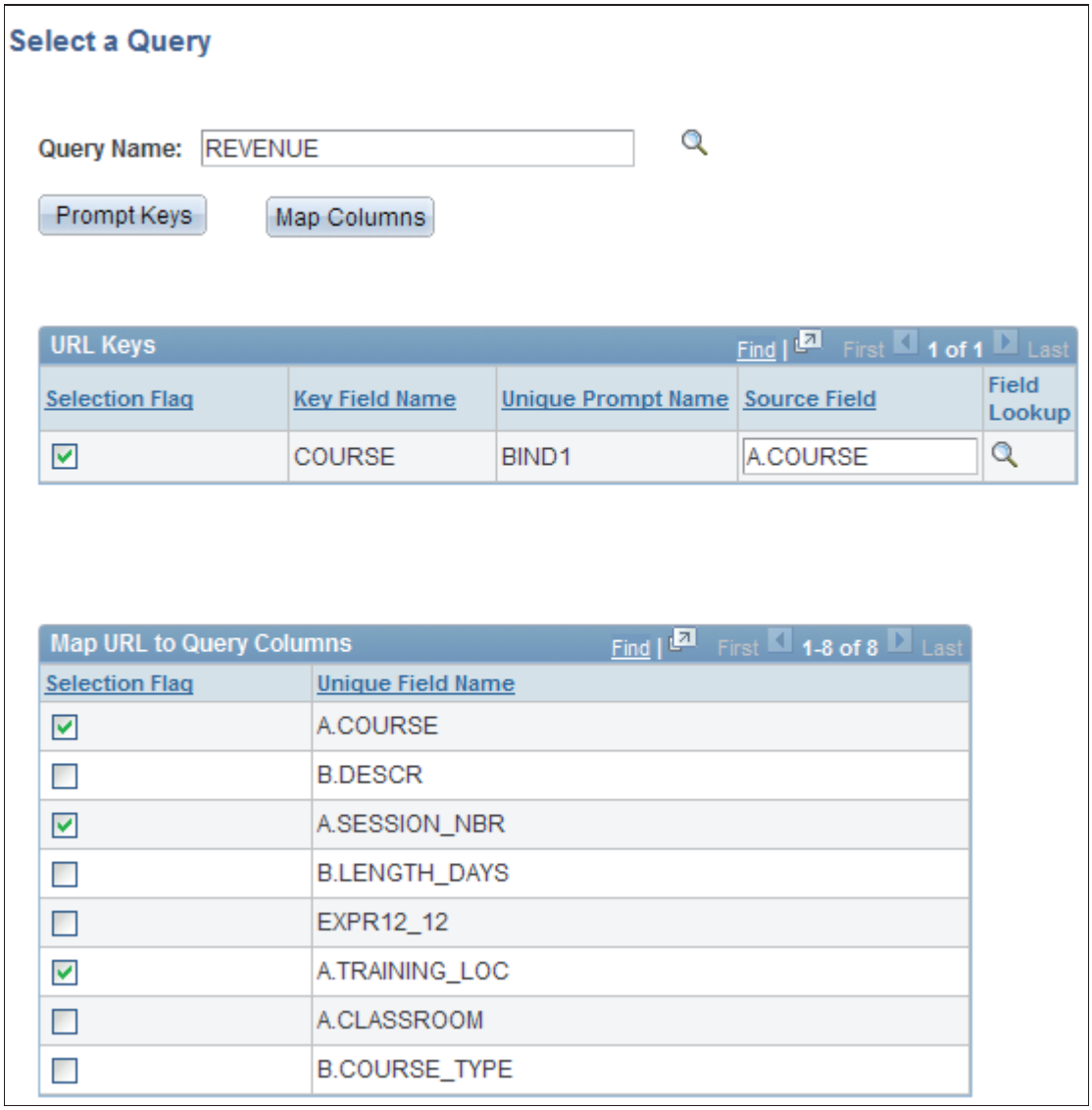
Defining Query URL Widgets for an Existing Query

To define query URL widgets for an existing query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Access Query Manager, and open the CLS001 query.
3. Save the query as *CLS001_DRILLING_URL*.
4. Select the Expressions tab, and click the Add Expression button.
5. Select the *Drilling URL* option from the Expression Type drop-down list box.
6. Click the Query URL link.
7. In the Query Name field, search for and select the *REVENUE* query.
8. Click the Prompt Keys button to bind the prompt keys of the REVENUE query with the columns of the CLS001_DRILLING_URL query.
9. Select the COURSE field as a prompt key field.
10. Click the Source Field lookup icon associated with the COURSE field, and select the A.COURSE field.
11. Click the Map Columns button.
12. Select the A.COURSE, A.SESSION_NBR, and A.TRAINING_LOC fields from the Map URL to Query Columns section.
13. Click the OK button twice to return to the Expressions page, and save the query.

Results

This example shows the settings in the Select a Query page:



Viewing Drilling URL Results

To view drilling URL results:

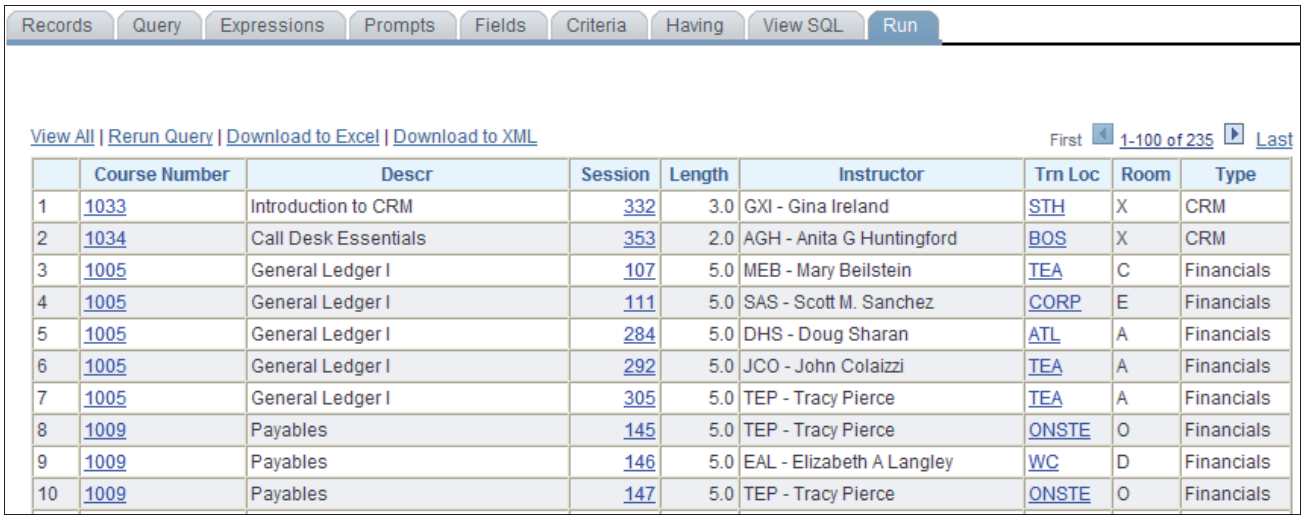
1. From the Expressions page, select the Run tab.
2. Click the 1002 link in the Course Number column.
3. Enter the following information for the REVENUE prompts:

Page Element	***Value or Status***
Enter Course	*1002*
Enter Enroll Start Date	*07/01/2009*
Enter Enroll End Date	*12/31/2009*

4. Click the View Results button, and compare the reports with these results.

Results

This is the CLS001_DRILLING_URL query with 235 rows of results displayed as links:

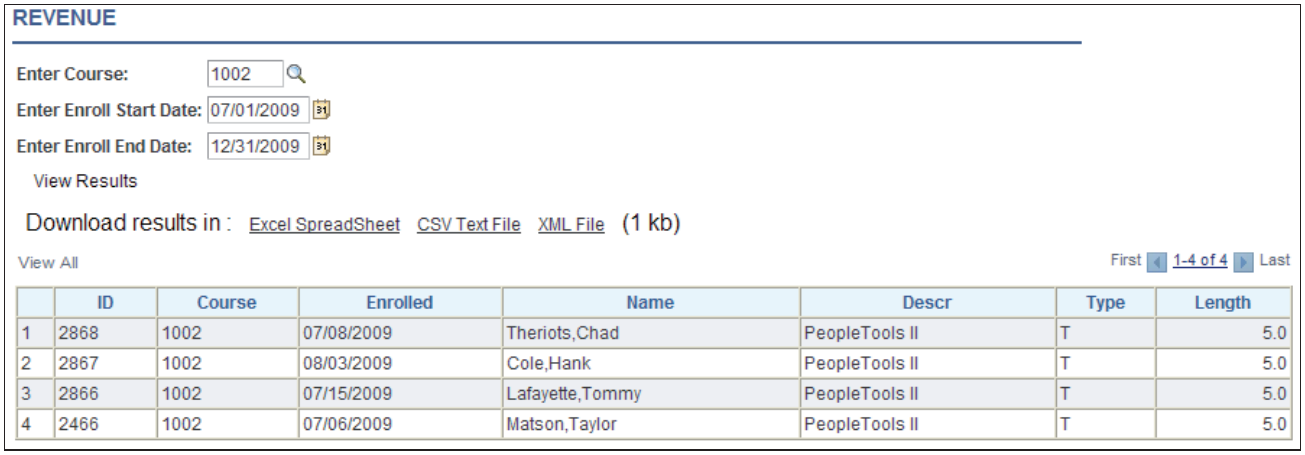


Records | Query | Expressions | Prompts | Fields | Criteria | Having | View SQL | Run

View All | Rerun Query | Download to Excel | Download to XML — First 1-100 of 235 Last

	Course Number	Descr	Session	Length	Instructor	Trn Loc	Room	Type
1	1033	Introduction to CRM	332	3.0	GXI - Gina Ireland	STH	X	CRM
2	1034	Call Desk Essentials	353	2.0	AGH - Anita G Huntingford	BOS	X	CRM
3	1005	General Ledger I	107	5.0	MEB - Mary Beilstein	TEA	C	Financials
4	1005	General Ledger I	111	5.0	SAS - Scott M. Sanchez	CORP	E	Financials
5	1005	General Ledger I	284	5.0	DHS - Doug Sharan	ATL	A	Financials
6	1005	General Ledger I	292	5.0	JCO - John Colaizzi	TEA	A	Financials
7	1005	General Ledger I	305	5.0	TEP - Tracy Pierce	TEA	A	Financials
8	1009	Payables	145	5.0	TEP - Tracy Pierce	ONSTE	O	Financials
9	1009	Payables	146	5.0	EAL - Elizabeth A Langley	WC	D	Financials
10	1009	Payables	147	5.0	TEP - Tracy Pierce	ONSTE	O	Financials

This example shows the REVENUE query displaying the enrollment dates of course 1002, between 07/01/2009 and 12/31/2009:



REVENUE

Enter Course: 1002

Enter Enroll Start Date: 07/01/2009

Enter Enroll End Date: 12/31/2009

View Results

Download results in : Excel SpreadSheet CSV Text File XML File (1 kb)

View All — First 1-4 of 4 Last

	ID	Course	Enrolled	Name	Descr	Type	Length
1	2868	1002	07/08/2009	Theriots,Chad	PeopleTools II	T	5.0
2	2867	1002	08/03/2009	Cole,Hank	PeopleTools II	T	5.0
3	2866	1002	07/15/2009	Lafayette,Tommy	PeopleTools II	T	5.0
4	2466	1002	07/06/2009	Matson,Taylor	PeopleTools II	T	5.0

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- Drilling URLs are the URLs that you define by selecting the menu, component, page, portal object, or URL of choice.
- Drilling URLs are a special type of expression that you can define using the Edit Expression Properties page in Query Manager.
- When you access the Run page of a query that has a Drilling URL defined, its query results are shown as links.
- You can schedule queries with drilling URLs using the Schedule Query page, which has five different output types: HTM, PDF, TXT, XLS, and XML.

Slide 160

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Drilling URL in Oracle PeopleSoft Query	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,* "Defining Selection Criteria"

Lesson 12

Implementing Any Joins

Objectives

By the end of this lesson, you will be able to:

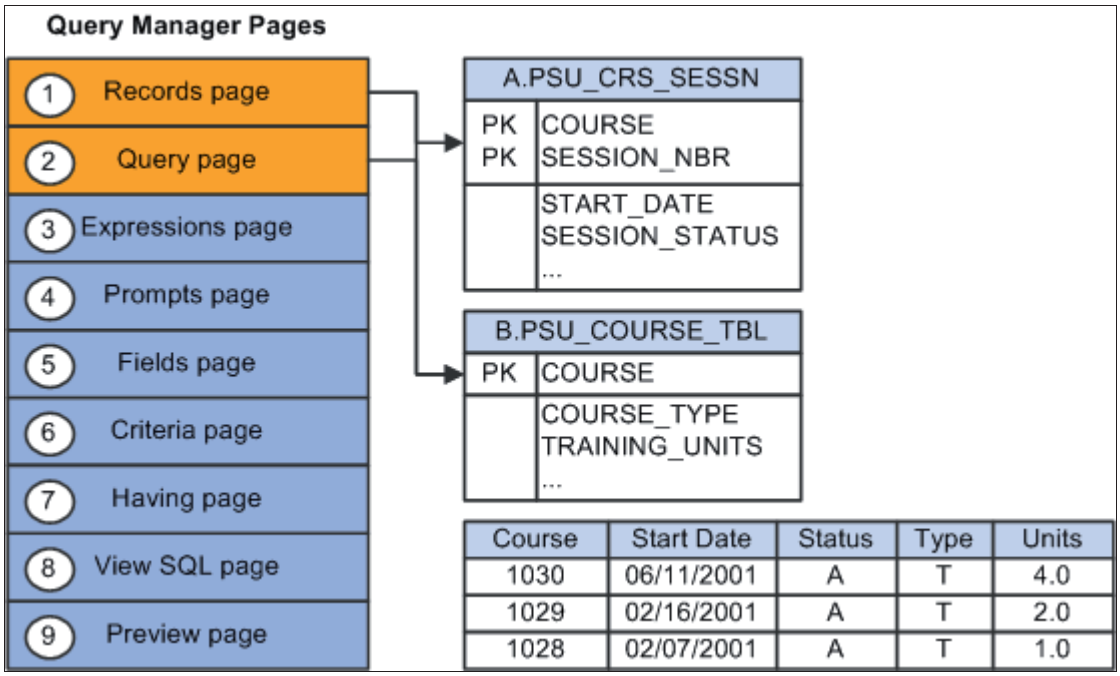
- Explain Any joins.
- Create Any joins.

Slide 162

Explaining Any-Joins

Purpose of Any-Joins

You can create queries based on multiple tables, even when a table is not in the parent hierarchy or related-record hierarchy. This is an example of an any-join:



Slide 163

Student Notes

Joining Tables with Any Joins

Any joins are manually linked to tables to retrieve the correct output. The tables are linked using common keys.

For example, the Student Data (PSU_STUDENT_TBL) and Customer (PSU_CUST_TBL) tables include these fields:

PSU_STUDENT_TBL	***PSU_CUST_TBL***
STUDENT_ID (Key field) NAME CUSTOMER_ID	CUSTOMER_ID (Key field) DESCR

If you want to retrieve information about students and their companies, you need to pull data from both the Student Data table and the Customer table.

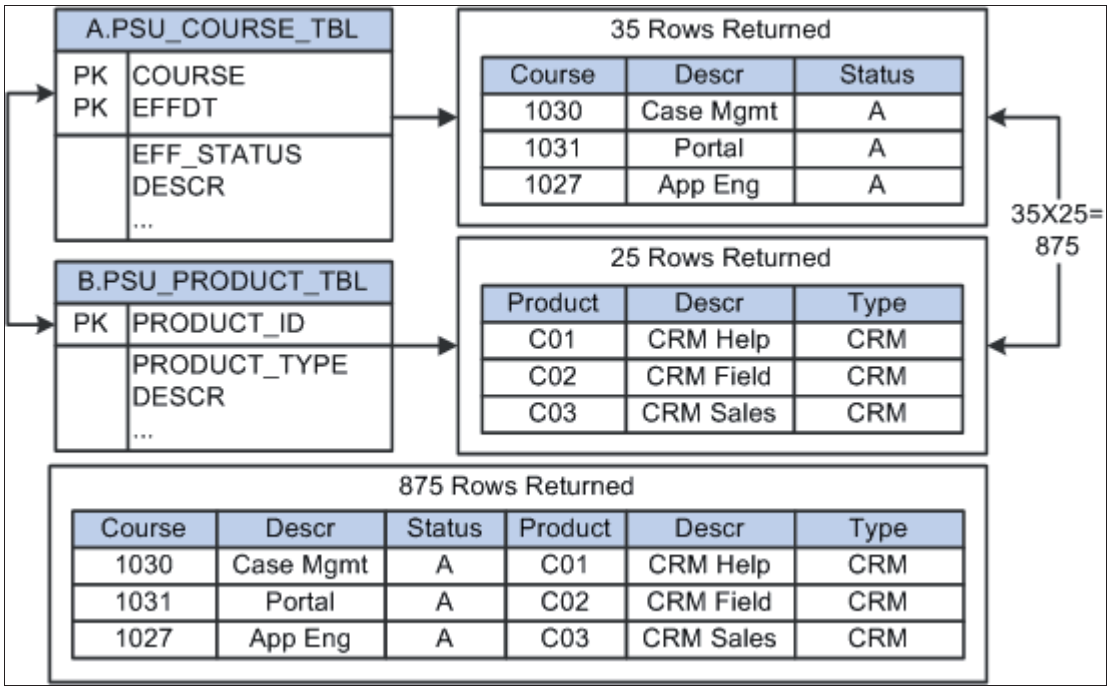
This SQL shows the WHERE clause making the join with the common key CUSTOMER_ID:

```
SELECT  A.STUDENT_ID,
         A.NAME,
         B.DESCR
FROM  PSU_STUDENT_TBL A,
          PSU_CUST_TBL B
WHERE  A.CUSTOMER_ID =
         B.CUSTOMER_ID
```

Explaining Any-Joins (continued)

Identifying Cartesian Joins

This example shows that each of the 35 rows in the Course table is joined to all 25 rows in the Product table, retrieving a total of 875 total rows:



Slide 164

Student Notes

Identifying Cartesian Joins

Cartesian joins match each row in the A table to every row in the B table. You create a Cartesian join when the WHERE clause is incorrect and the SQL cannot resolve the joining criteria.

Problem with Cartesian Joins

Cartesian joins produce output data that is essentially useless, unless you are doing volume testing on a server.

Cartesian joins decrease database performance when the two tables contain large numbers of rows.

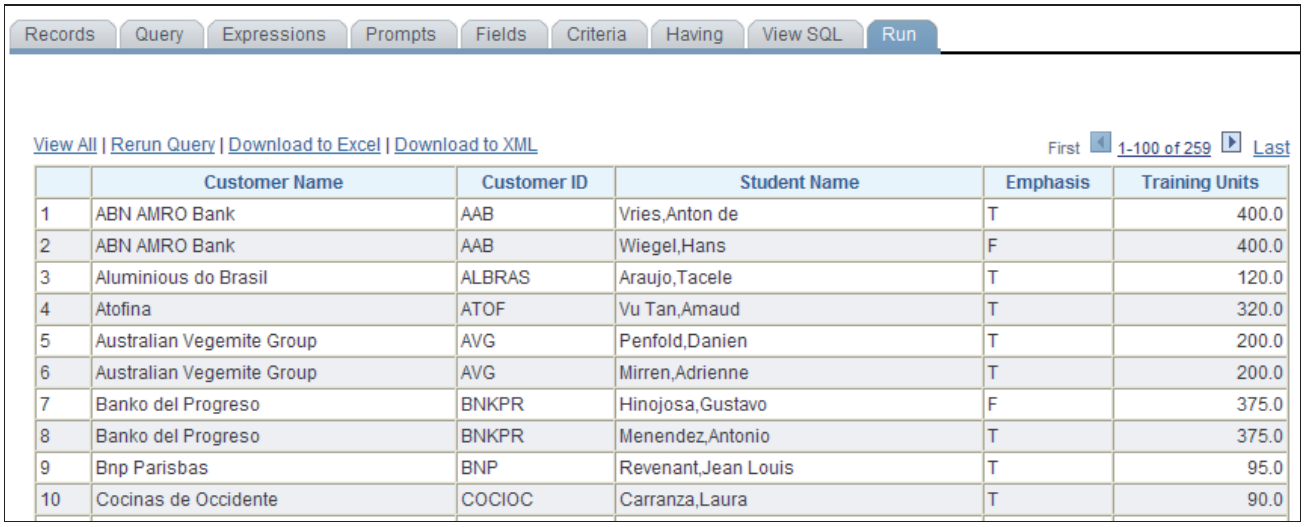
Using Any Joins

Use Any joins when:

- You require additional information, but no one record stores the data.
- No predefined joins include the record that you need.

The query that is shown in the following example uses the Student Data record to retrieve information. The query also needs to retrieve specific data about the customer.

The Student Data record does not contain many fields regarding the customers. Join the Student Data record with the Customer record using the CUSTOMER_ID field as the common field from both records. The query retrieves the necessary information, as shown:



	Customer Name	Customer ID	Student Name	Emphasis	Training Units
1	ABN AMRO Bank	AAB	Vries,Anton de	T	400.0
2	ABN AMRO Bank	AAB	Wiegel,Hans	F	400.0
3	Aluminious do Brasil	ALBRAS	Araujo,Tacele	T	120.0
4	Atofina	ATOF	Vu Tan,Amaud	T	320.0
5	Australian Vegemite Group	AVG	Penfold,Danien	T	200.0
6	Australian Vegemite Group	AVG	Mirren,Adrienne	T	200.0
7	Banko del Progreso	BNKPR	Hinojosa,Gustavo	F	375.0
8	Banko del Progreso	BNKPR	Menendez,Antonio	T	375.0
9	Bnp Parisbas	BNP	Revenant,Jean Louis	T	95.0
10	Cocinas de Occidente	COCIOC	Carranza,Laura	T	90.0

Example: Any-Join SQL

The Student Data record and Customer records include these fields:

PSU_STUDENT_TBL	***PSU_CUST_TBL***
STUDENT_NAME (key field)	CUSTOMER_ID (key field)
CUSTOMER_ID	DESCR
EMPHASIS	TRAINING_UNITS

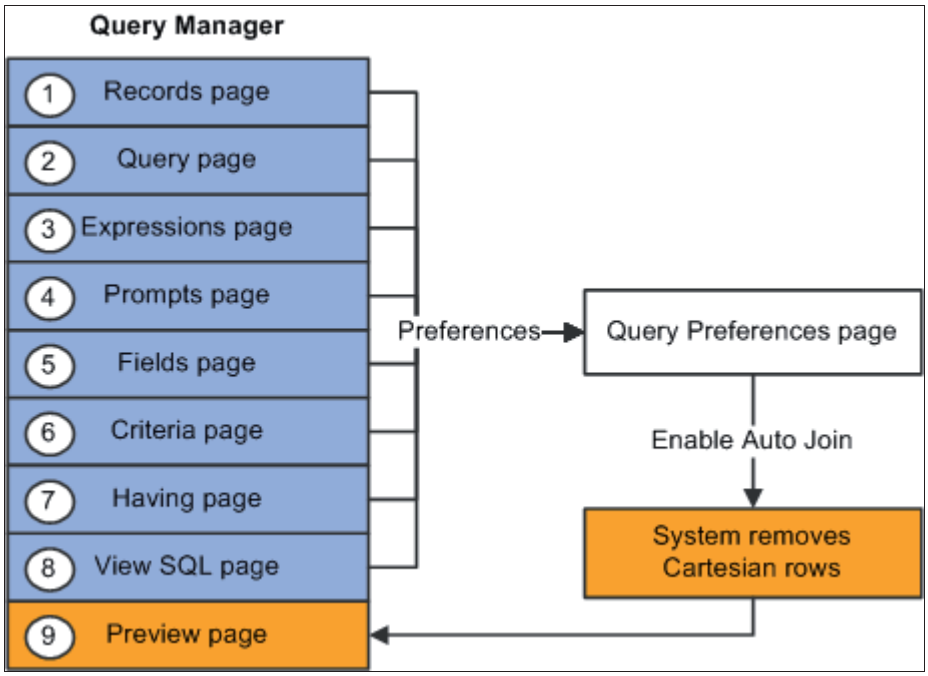
Join the two records on the CUSTOMER_ID field, which the two records have in common, as in this SQL:

```
SELECT  A.STUDENT_NAME, A.CUSTOMER_ID, A.EMPHASIS,
            B.DESCR, B.TRAINING_UNITS
         FROM  PS_PSU_STUDENT_TBL A, PS_PSU_CUST_TBL B
         WHERE  A.CUSTOMER_ID = B.CUSTOMER_ID
```

Creating Any Joins

Using the Auto-Join Feature

This diagram shows the path to the Auto-join feature that creates a row of criteria that joins two records with matching key fields.



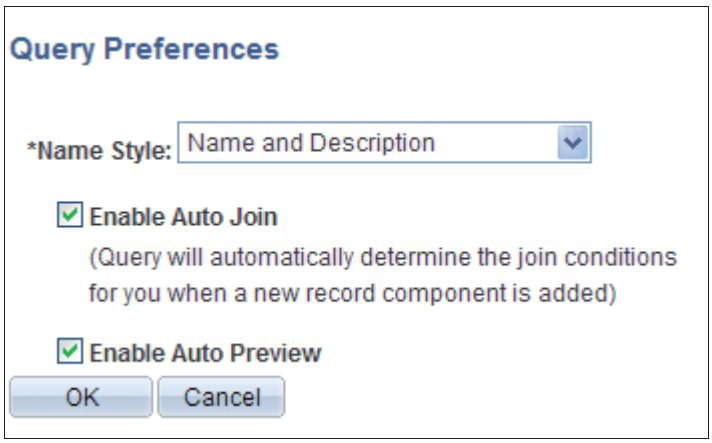
Slide 165

Student Notes

Page Used to Enable the Auto-Join Feature

Page Name	*Navigation*
Query Preferences	1. Select Reporting Tools, Query, Query Manager. 2. Create a new query or open an existing one. 3. In Query Manager pages, click the Preferences link.

Use this page to enable the Auto-join feature:



The Auto-Join Feature

When you enable the Auto-join feature, PeopleSoft Query determines the join conditions and prompts you to add the join to WHERE clause in the criteria.

The default setting for the Auto-join feature is enabled. You can disable it by using the Preferences link in Query Manager.

Note. The Auto-join feature performs the join on all matching fields with the exception of the EFFDT field and the EFF_SEQ field.

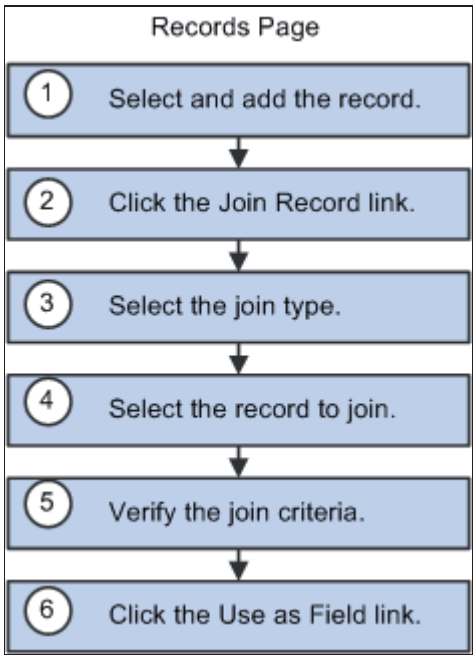
The Auto Preview Feature

When you enable the Auto Preview feature, the application refreshes the data when you select the Run page. When you disable this feature, you refresh the data manually by clicking the Rerun Query link on the Run page.

Creating Any Joins (continued)

Steps for Creating an Any-Join

Follow these steps to use an any-join (the auto-join feature):



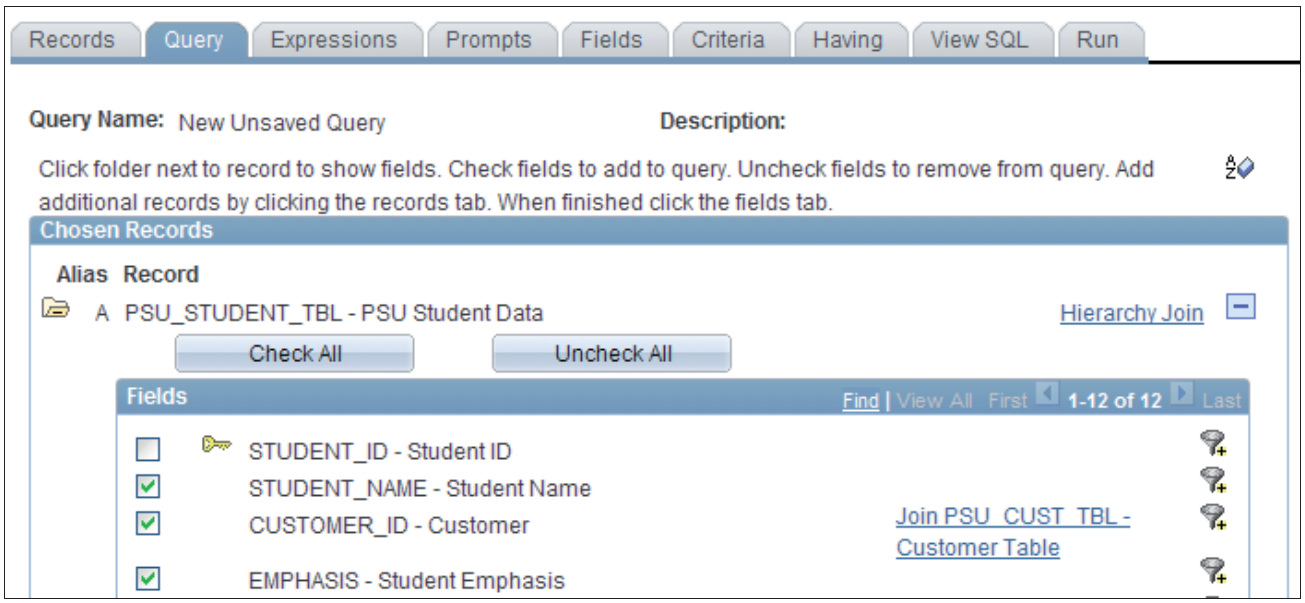
Slide 166

Student Notes

Procedure to Create an Any-Join

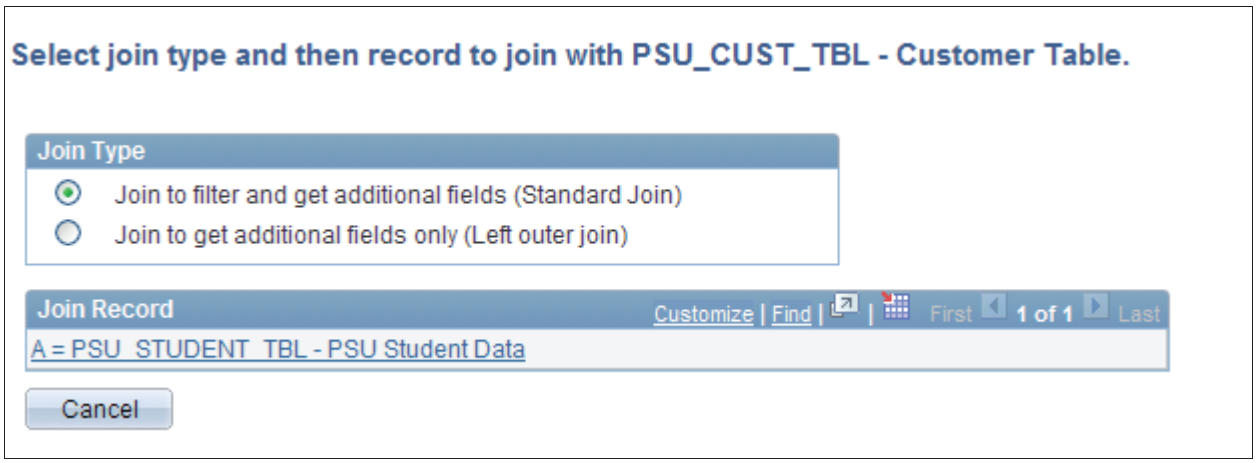
The following example shows a query with one record. The second record must have a key field that is also a field in the first record to perform an any-join. On the Query page, the key symbol identifies key fields.

In this example, the CUSTOMER_ID field is used to join the Customer table:

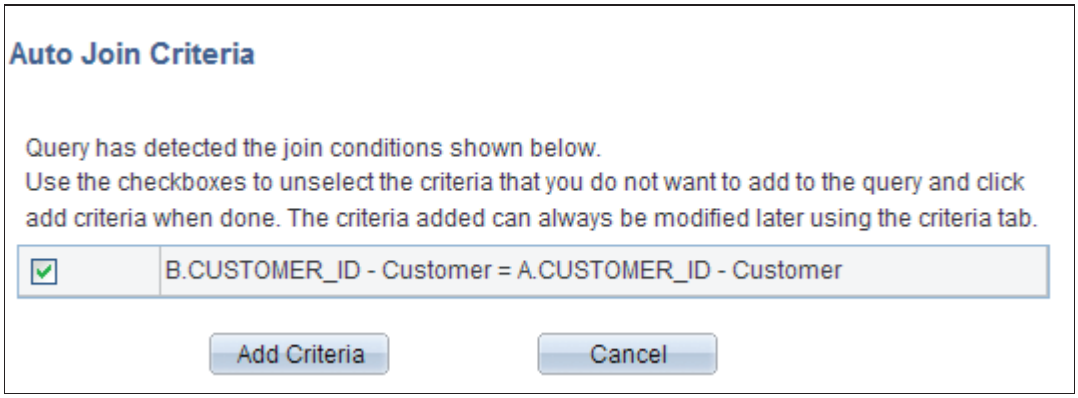


Use the Records page to search for the record that is used for the join, and then click the Join Record link:

Select the join type to join PSU_CUST_TBL to PSU_STUDENT_TBL, and click the link in the Join Record list, as shown here:



Verify that the join criteria is correct, and click the Add Criteria button:



Activity 22: Joining Multiple Records

In this activity, you will review the activity overview and:

1. Create a query.
2. Join records.

Slide 167

Activity Overview

In this activity, you create the CUSTOMER_ANALYSIS query. The query retrieves student, customer, and product information. Display student name, customer ID, curriculum emphasis, product description, database platform, skill, and proficiency level. Sort the output alphabetically by customer.

The student data (PSU_STUDENT_TBL) record stores the student name, curriculum emphasis, and customer ID values. The student experiences (PSU_STUDENT_EXP) record stores the skill and proficiency data. The customer product table (PSU_CUST_PROD) contains the products that the customer has purchased and the customer's database platform.

Preview the query, and compare the results to the activity results.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Use the following information to create a query:

Record	***Fields***
PSU_STUDENT_TBL (table A)	STUDENT_NAME (RFT Long) CUSTOMER_ID (order by 1) EMPHASIS (XLAT Short)

3. Save the query as CUSTOMER_ANALYSIS.

Joining Records

To join records:

1. Select the Records tab, and search for the PSU_CUST_PROD record.
2. Click the Join Record link.
3. Accept the default for the join type, and then click the A=PSU_STUDENT_TBL-PSU Student Data link.
4. Click the Add Criteria button.
5. Use the following information to select the fields in the customer product record:

Record	***Fields***
PSU_CUST_PROD (table B)	PRODUCT PLATFORM (XLAT Short)

6. Save the query.
7. Select the Records tab, and search for the PSU_STUDENT_EXP record.
8. Click the Join Record link.
9. Accept the default for the join type.
10. Click the A=PSU_STUDENT_TBL-PSU Student Data link.
11. Click the Add Criteria button.

12. Use the following information to select the fields for the first any-join:

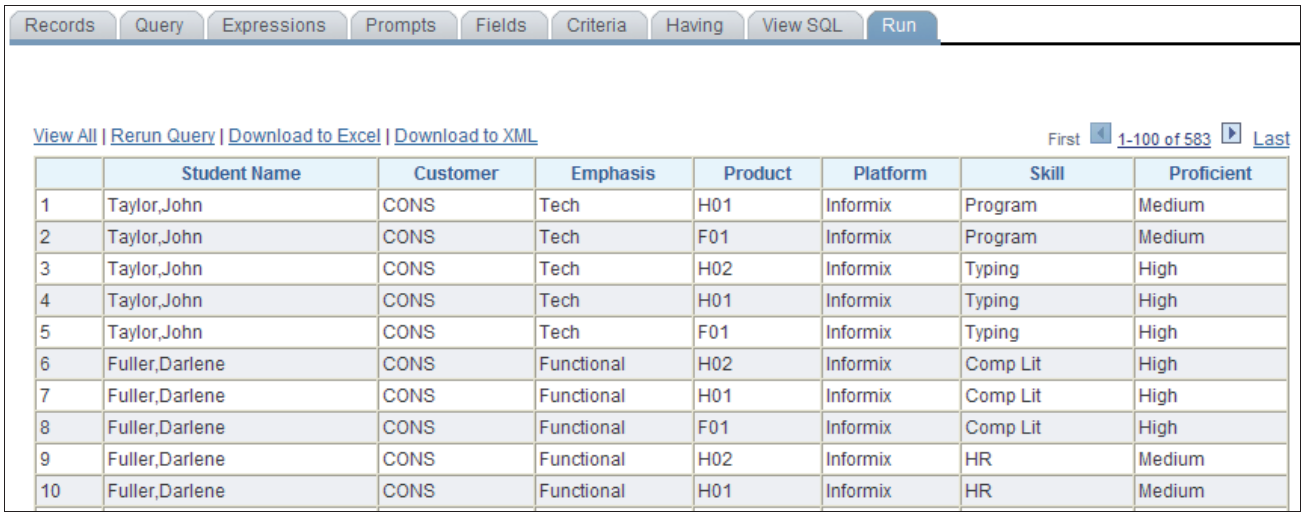
Record	***Fields***
PSU_STUDENT_EXP (table C)	SKILL (XLAT Short) PROFICIENCY (XLAT Short)

13. Save and preview the query.

14. Compare the output with the following results.

Results

The CUSTOMER_ANALYSIS query returns 583 rows:



	Student Name	Customer	Emphasis	Product	Platform	Skill	Proficient
1	Taylor,John	CONS	Tech	H01	Informix	Program	Medium
2	Taylor,John	CONS	Tech	F01	Informix	Program	Medium
3	Taylor,John	CONS	Tech	H02	Informix	Typing	High
4	Taylor,John	CONS	Tech	H01	Informix	Typing	High
5	Taylor,John	CONS	Tech	F01	Informix	Typing	High
6	Fuller,Darlene	CONS	Functional	H02	Informix	Comp Lit	High
7	Fuller,Darlene	CONS	Functional	H01	Informix	Comp Lit	High
8	Fuller,Darlene	CONS	Functional	F01	Informix	Comp Lit	High
9	Fuller,Darlene	CONS	Functional	H02	Informix	HR	Medium
10	Fuller,Darlene	CONS	Functional	H01	Informix	HR	Medium

This concludes the activity. Please do not continue.

Activity 23: Using Advanced Selection Criteria

In this activity, you will review the activity overview and:

1. Create a query.
2. Create a related-record join.
3. Create the any-join.
4. Create date range prompts.

Slide 168

Activity Overview

Create the COURSE_START_DT query. The Training department wants to see the number of course sessions that are available between July 01, 2009, and September 30, 2009. Display the Training location description and country from the Training location record (PSU_TRNLOC_TBL). Display the name of each location state name as stored in the state record (STATE_TBL). Display the total count of the course sessions and the session start date from the course session record (PSU_CRS_SESSN). The Training department asks that you present the information in chronological order and with descriptive headings.

Preview the query, and compare your results to the activity results.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Access Query Manager, and use the following information to create the query:

Record	*Fields (Properties)*
PSU_TRNLOC_TBL	DESCR (Training Location) COUNTRY (RFT Long)

Note. Click the OK button to accept the Effective Date criteria.

3. Save the query as COURSE_START_DT.

Creating a Related-Record Join

To create a related-record join:

1. Select the Query tab, and click the join link to the right of the STATE field.
2. Select the default standard join, and click the OK button.
3. Use the following information to select the fields:

Record	*Field (Properties)*
STATE_TBL (B)	DESCR (State)

4. Save the query.

Creating the Any-Join

To create the any-join:

1. Select the Records tab, and search for the PSU_CRS_SESSN record.
2. Click the Join Record link for the PSU_CRS_SESSN record.
3. Accept the default standard join type, and join the record to the PSU_TRNLOC_TBL.
4. Click the Add Criteria button.

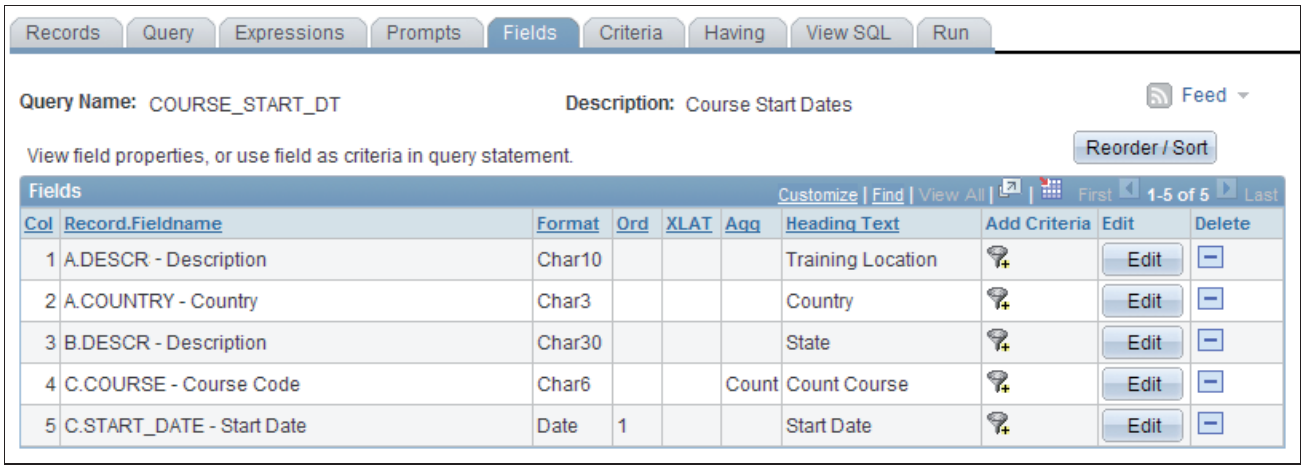
5. Use the following information to select the fields:

Record	*Fields (Properties)*
PSU_CRS_SESSN (C)	COURSE (Heading: *Count Course*) (Aggregate: *Count*) START_DT (RFT Long, Order by 1)

6. Save the query, and compare the Fields page with the following example.

Results

This example shows the Fields page after you created a related-record join:



Creating Date Range Prompts

To create date range prompts:

1. In the Fields page, click the Add Criteria icon next to the START_DATE field.
2. Select the *Prompt* option under the Choose Expression 2 Type section.
3. Click the New Prompt link under the Expression 2 section, and enter the following information:

Page Element	*Value or Status*
Heading Type	*Text*
Heading Text	*Course Start Date Between*
Edit Type	*No Table Edit*

4. Click the OK button to return to the Edit Criteria Properties page.

5. Click the New Prompt link under the Expression 2 section, and enter the following information:

Page Element	***Value or Status***
Heading Type	*Text*
Heading Text	*and*
Edit Type	*No Table Edit*

6. Click the OK button, and enter the following information:

Page Element	***Value or Status***
Condition Type	*between*
Choose Expression 2 Type	*Expr - Expr*

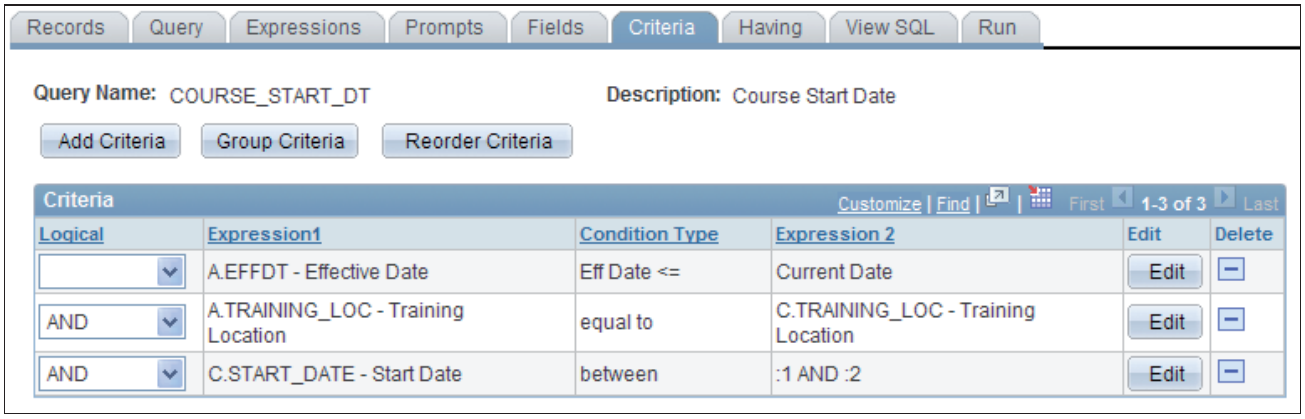
7. In the first Define Expression group box, click the Add Prompt link.
8. Select the *Course Start Date Between* prompt that you just set up.
9. In the second Define Expression group box, click the Add Prompt link.
10. Select the *and* prompt that you just set up.
11. Click the OK button, and save your query.
12. Preview the Criteria and the Prompts page.
13. Preview the query results using these values:

Page Element	***Value or Status***
Course Start Date Between	*07/01/2009*
and	*09/30/2009*

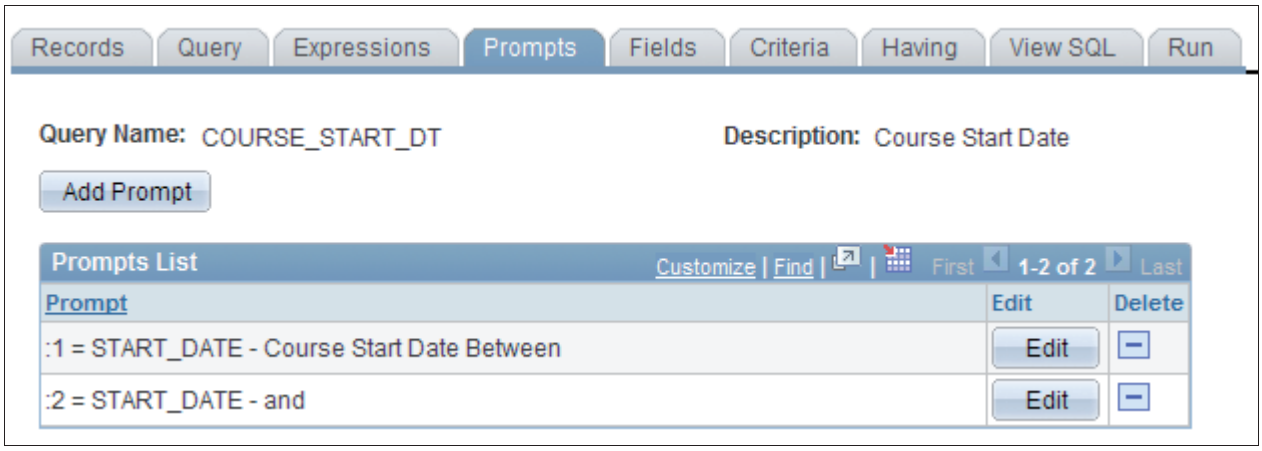
14. Compare the output with the following results.

Results

The Criteria page after you created the date range prompts:



The Prompts page after you created the date range prompts:



The COURSE_START_DT query returns six rows of results:

	Training Location	Country	State	Count Course	Start Date
1	Atlanta Training Center	USA	Georgia	1	07/27/2009
2	Corporate Headquarters	USA	California	1	07/28/2009
3	Walnut Creek Training Center	USA	California	1	08/17/2009
4	Corporate Headquarters	USA	California	1	08/24/2009
5	Pleasanton	USA	California	1	08/24/2009
6	Bethesda Training Center	USA	Maryland	1	08/24/2009

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- Any-joins enable you to join tables when hierarchy and related-record joins are not available. You establish any-joins by linking tables with common key fields.
- You create any-joins by using the auto-join feature, which you enable on the Query Preferences page in Query Manager.

Slide 169

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Explaining Any joins	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Creating and Running Simple Queries"
Using aggregates in expressions Using prompts in expressions	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Defining Selection Criteria"

Lesson 13

Using Subqueries

Objectives

By the end of this lesson, you will be able to:

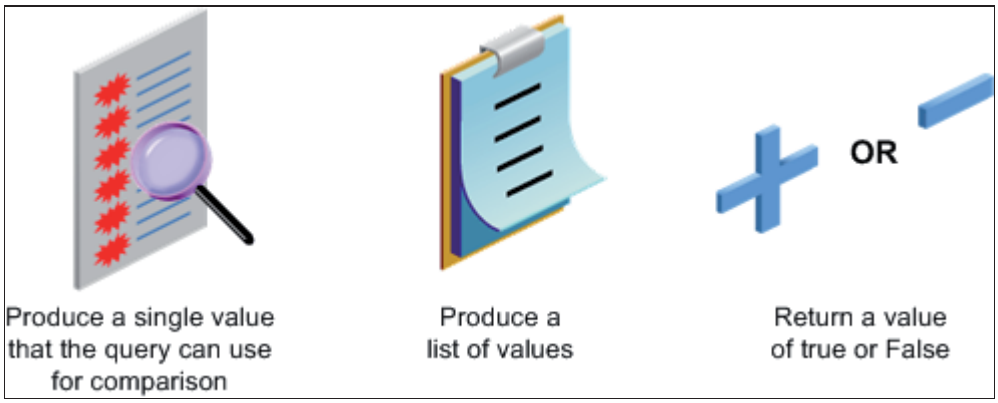
- Explain subqueries.
- Create a single-value subquery.
- Create an in-list or not-in-list subquery.
- Create an exist or does-not-exist subquery.

Slide 171

Explaining Subqueries

Purpose of Subqueries

A subquery is a query within a query. This diagram lists uses of subqueries:



Slide 172

Student Notes

Purpose of Subqueries

You use subqueries to compare the value for a field in the primary query to the results of a subordinate query. You embed the subordinate query in the WHERE clause using the Criteria page.

Note these points about subqueries:

- The condition type that you specify in the criteria determines what the subquery returns to the query.
- A subquery can retrieve only one data field from a single table, and the subquery can contain a join.

  You can use this feature to specify criteria based on two records.
- The system never displays the result of the subquery; it displays the results of the query, and they are *limited by* the subquery.
- Additional rows of criteria can be placed in the primary query or the subquery.
- To set up a subquery, select the Criteria page, select Subquery as the expression type for Expression 2, and then click the Define/Edit Subquery link.

  After clicking the Define/Edit Subquery link, the Records page appears, and you can select a record for the subquery definition.

Creating a Single-Value Subquery

Purpose of Single-Value Subqueries

A single-value subquery enables you to compare detail values to an aggregate value.

The following SQL example demonstrates a subquery:

```
SELECT  A.CUST ID,
        A.TRAINING_UNIT
FROM  PS_PSU_CUST_TBL A
WHERE   A.TRAINING_UNIT >
          (SELECT AVG(B.TRAINING_UNIT) FROM PS_PSU_CUST_TBL B)
```

Slide 173

Student Notes

Example: Subquery

A subquery is a query within a query. By creating nested queries, you have the first query display the fields necessary for the results and the subordinate query filter the data.

Because salary information and personnel rosters change frequently, a query that contains a hard-coded value for an average salary can quickly become outdated. If you use a subquery, you can recalculate that average with each run, which enables you to include current information in the query.

For example, suppose that the average training unit cost for customers is 40.40 USD.

In the previous lessons, you learned that an aggregated field loses all individual values; therefore, you could not create a query with the following criterion:

```
TRAINING_UNIT > AVG(TRAINING_UNIT)
```

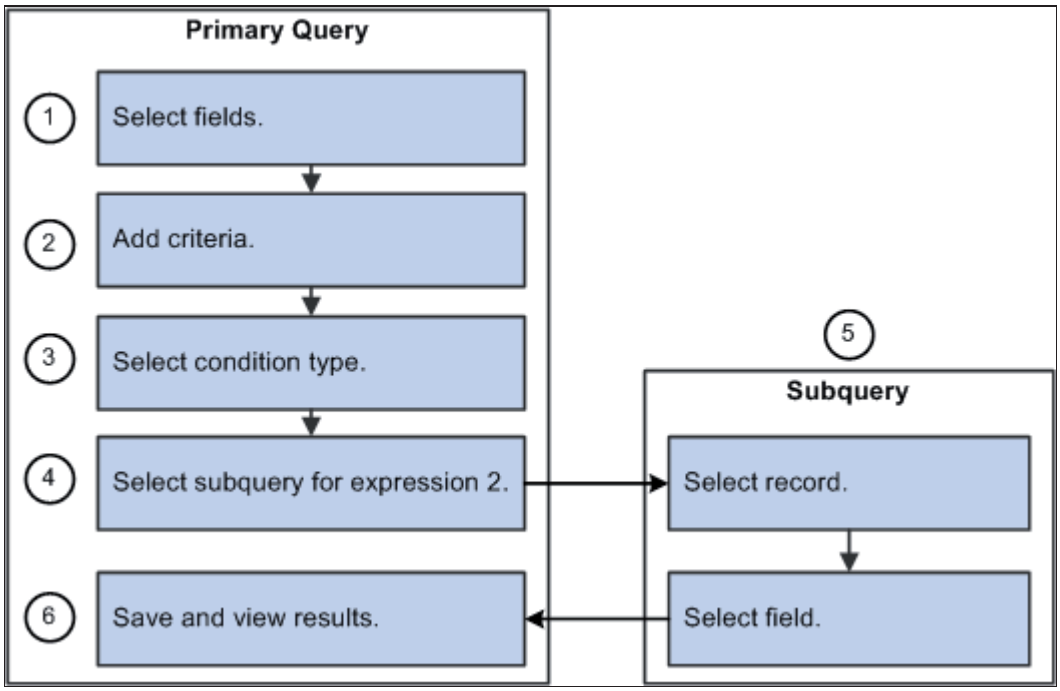
However, you could create a query where the subquery calculates the average of the TRAINING_UNIT field with the following criterion:

```
TRAINING_UNIT > Subquery
```

Creating a Single-Value Subquery (continued)

Steps for Creating a Single-Value Subquery

Use the following process to create a single-value subquery:



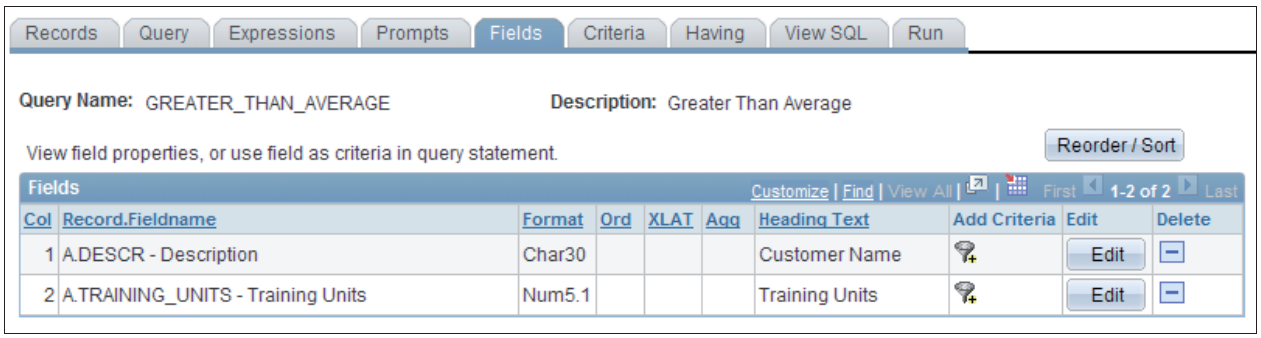
Slide 174

Student Notes

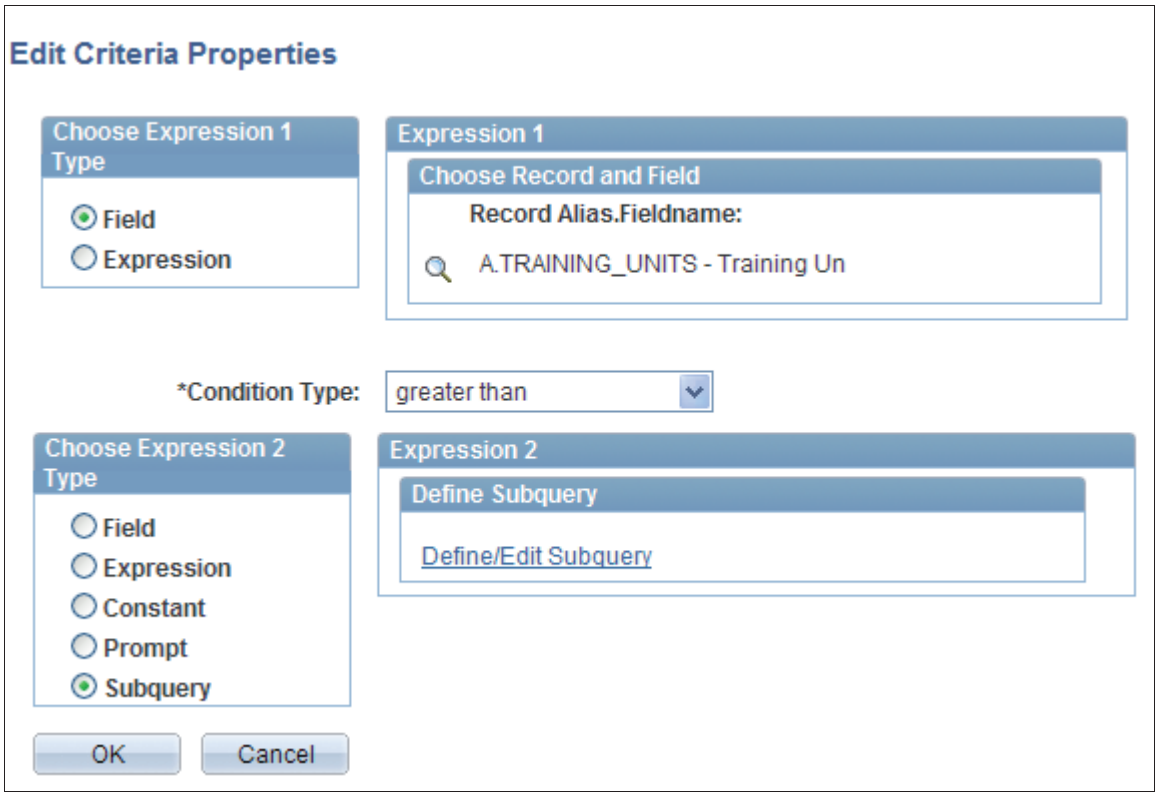
Creating Single-Value Subqueries

Use the following steps to create a single-value subquery:

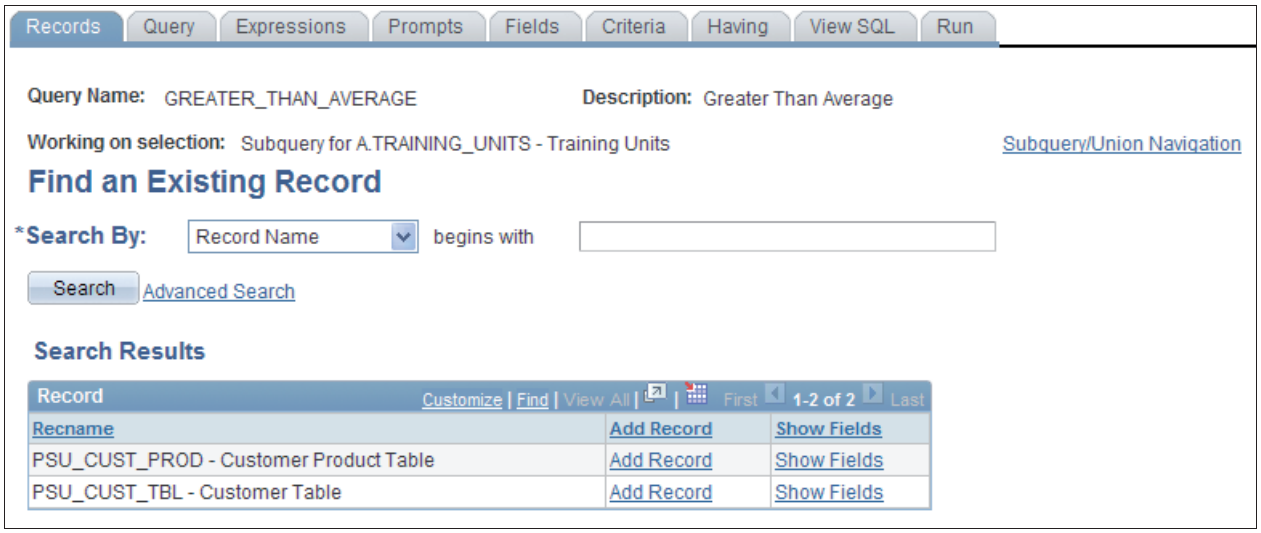
1. Select the fields to appear on the Run page from the first record; for example, PSU_CUST_TBL:



2. Add criteria to the first query to access the subquery:



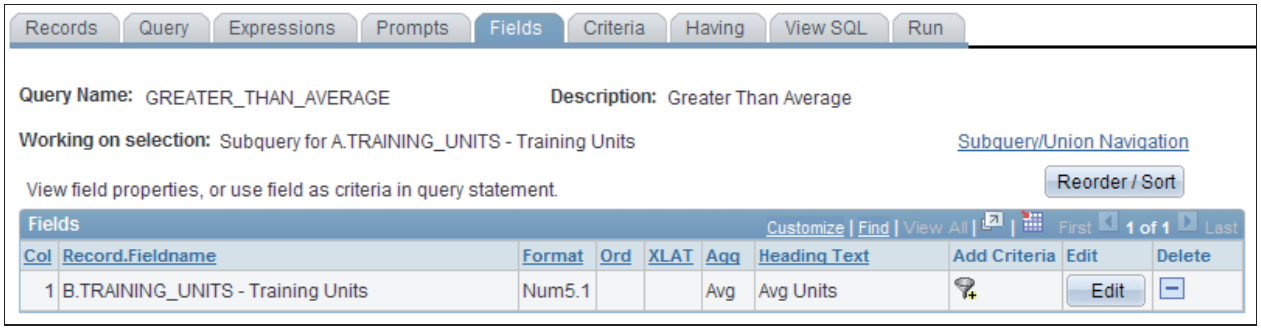
3. When you click the Define/Edit Subquery link, the Records page enables you to select the record for the subordinate query. (If necessary, you can use the same record as the first query.)



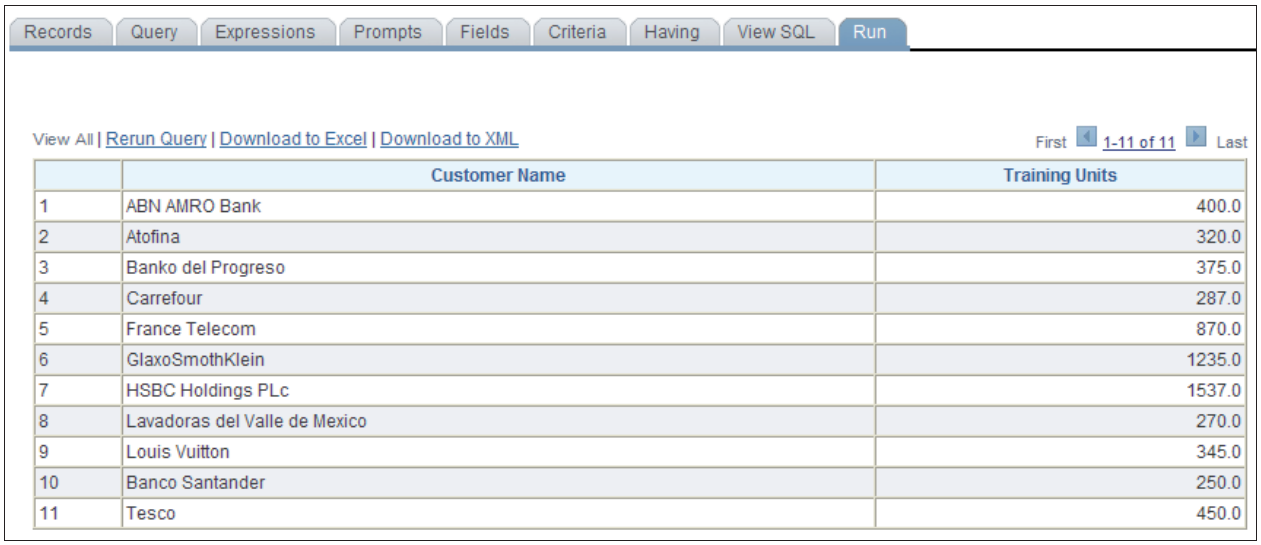
Note. The Working on selection field displays informational text to help you keep track of whether you are working on the subquery or the primary query.

4. Select the record and its field for the subquery.

   For example, the Average aggregate determines the average number of training units, so you need to click the Add Record link of the PSU_CUST_TBL record, select the TRAINING_UNITS field, and set the aggregate of the TRAINING_UNITS to *Average:*



5. The results show only those rows that are greater than the average training units overall:

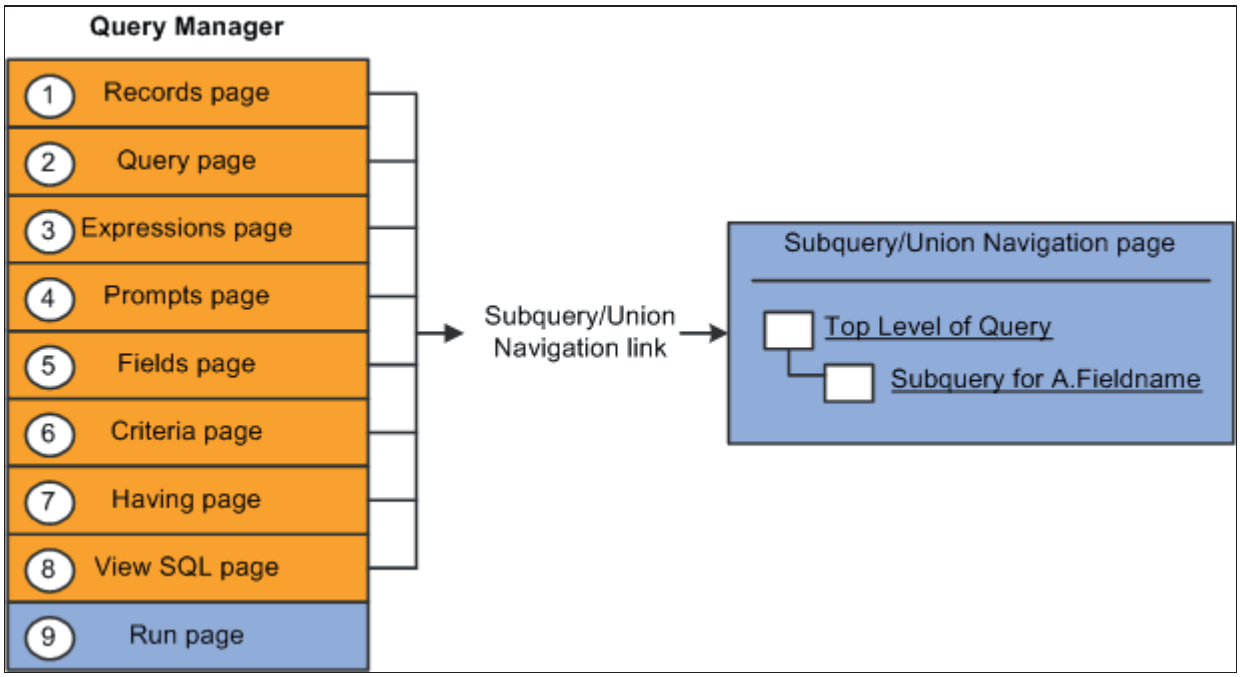


	Customer Name	Training Units
1	ABN AMRO Bank	400.0
2	Atofina	320.0
3	Banko del Progreso	375.0
4	Carrefour	287.0
5	France Telecom	870.0
6	GlaxoSmothKlein	1235.0
7	HSBC Holdings PLc	1537.0
8	Lavadoras del Valle de Mexico	270.0
9	Louis Vuitton	345.0
10	Banco Santander	250.0
11	Tesco	450.0

Creating a Single-Value Subquery (continued)

Subquery Navigation

You can use every page except the Run page to navigate between the top-level query and the subquery:



Slide 175

Student Notes

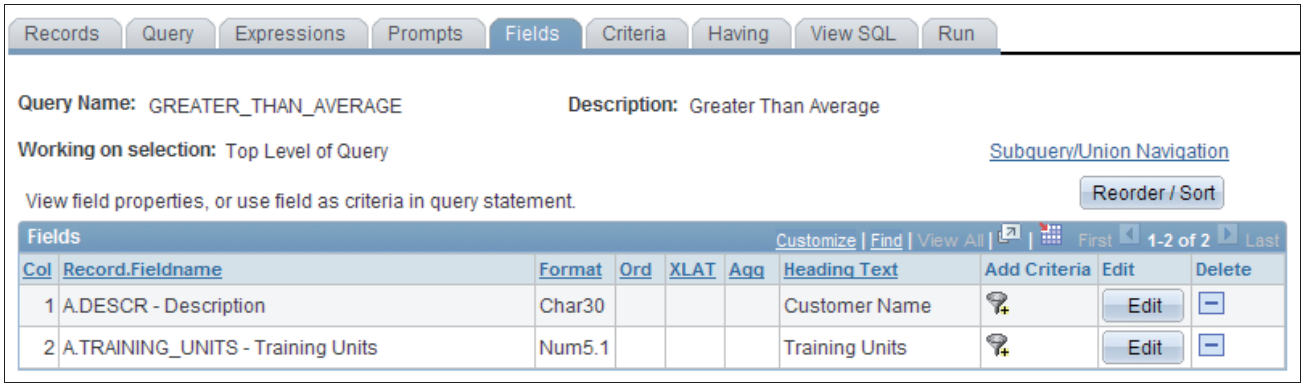
Navigating To and From a Subquery

After you create a subquery, you might need to navigate from the primary query to the subquery and then back to the primary query.

For example, you might need to:

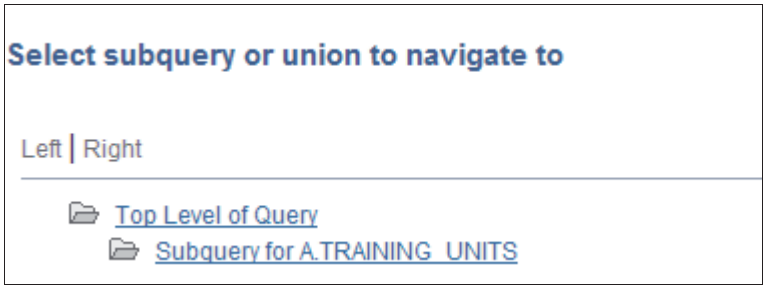
- Insert additional criteria in the primary query.
- Modify the primary query or the subquery.

You use the Subquery/Union Navigation link on the Fields page, as shown here:



Note. The Subquery/Union Navigation link appears on all pages except the Run page.

Click the link for the top-level query or for the subquery:



Activity 24: Creating Subqueries

In this activity, you will review the activity overview and:

- Create a query.
- Add a subquery as criteria.

Slide 176

Activity Overview

The Training department requests a query to find the customers who have more training units than the average number of training units for all customers. Display only the customer name and their training units. This information is stored in the Customer record (PSU_CUST_TBL).

Save the query as GREATER_THAN_AVERAGE.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Create a new query using the PSU_CUST_TBL record and the following fields:

Page Element	***Value or Status***
DESCR	Selected *Customer Name*
TRAINING_UNITS	Selected *Training Units*

3. Save the query as GREATER_THAN_AVERAGE.

Adding a Subquery as Criteria

To add a subquery as criteria:

1. Select the Criteria tab, and click the Add Criteria button.
2. Enter the following information on the Edit Criteria Properties page:

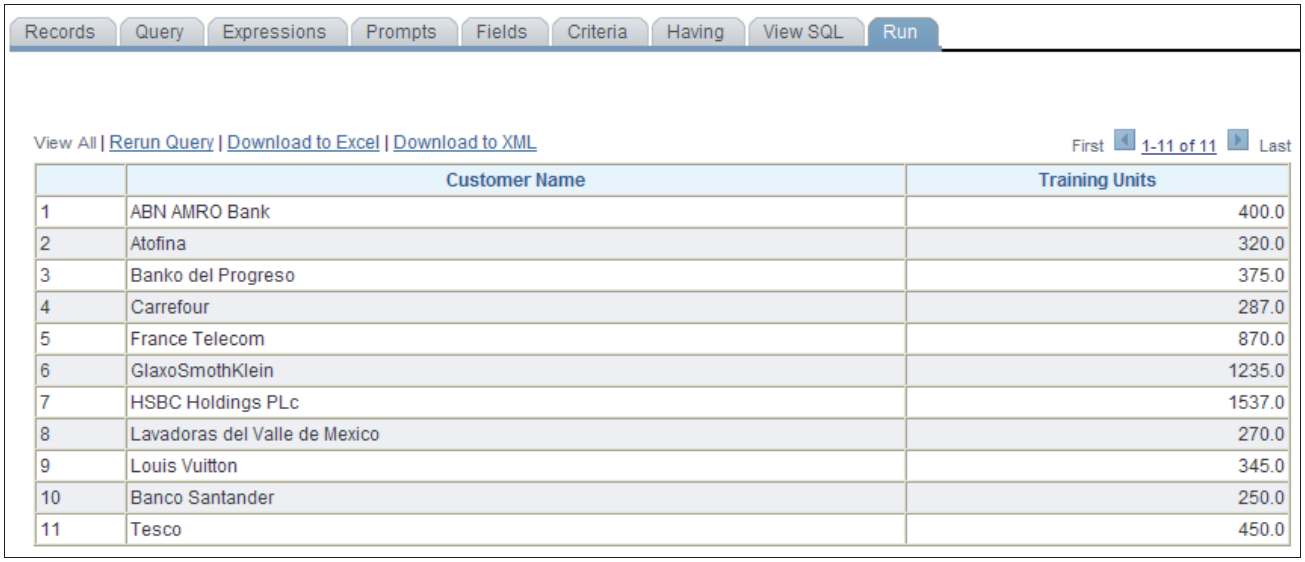
Page Element	***Value or Status***
Expression 1 Type	*Field*
Expression 1	*TRAINING_UNITS*
Condition Type	*greater than*
Expression 2 Type	*Subquery*

3. Click the Define/Edit Subquery link for Expression 2.
4. Click the Add Record link for the PSU_CUST_TBL table.
5. Click the Select link for the TRAINING_UNITS field.
6. Click the Edit button and then select the *Average* aggregate option.
7. Click the OK button.

8. Save and preview the query.
9. Compare the output with the following results.

Results

The GREATER_THAN_AVERAGE query returns 11 rows:



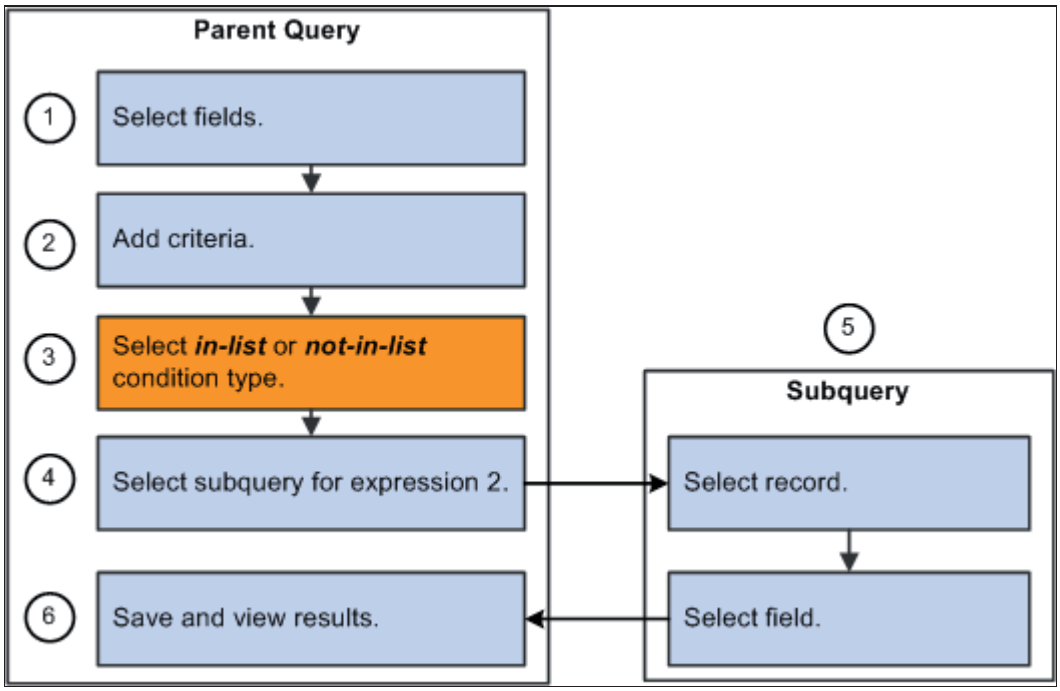
	Customer Name	Training Units
1	ABN AMRO Bank	400.0
2	Atofina	320.0
3	Banko del Progreso	375.0
4	Carrefour	287.0
5	France Telecom	870.0
6	GlaxoSmothKlein	1235.0
7	HSBC Holdings PLc	1537.0
8	Lavadoras del Valle de Mexico	270.0
9	Louis Vuitton	345.0
10	Banco Santander	250.0
11	Tesco	450.0

This concludes the activity. Please do not continue.

Creating an In-List or Not-In-List Subquery

Steps for Creating an In-List Subquery

Use these steps to create an *in-list* or *not-in-list* subquery:



Slide 177

Student Notes

In-List and Not-in-List Subqueries

When you select *in-list* or *not-in-list* as the condition type, subqueries retrieve data based on the criteria condition type from the primary query.

An *in-list* or *not-in-list* subquery returns the same data as an *exist* or *does-not-exist* subquery. Some database administrators believe that *exist* or *does-not-exist* subqueries are more efficient. Others find *in-list* or *not-in-list* subqueries to be simpler.

Here is the SQL code:

```
SELECT  A.STUDENT_ID,
          A.NAME
FROM  PS_PSU_STUDENT_TBL A
WHERE  NOT IN LIST
         (SELECT B.STUDENT_ID FROM PS_PSU_STU_ENROLL B)
```

This table explains the SQL in the example:

SQL	*Explanation*
SELECT A.STUDENT_ID, A.NAME FROM PS_PSU_STUDENT_TBL A	Retrieve the student ID and name columns from the student table.
WHERE NOT IN LIST	Retrieve only the values that are not in the following list.
(SELECT B.STUDENT_ID FROM PS_PSU_STU_ENROLL)	Retrieve a list of student IDs from the enrollment table.

Activity 25: Creating a Not-In-List Subquery

In this activity, you will review the activity overview and:

- Create a query.
- Add a subquery.

Slide 178

Activity Overview

The Training department needs to see the course codes and descriptions of courses that no student has taken or enrolled in. This information is in the Course (PSU_COURSE_TBL) and Student Enrollment (PSU_STU_ENROLL) records.

Save the query as *NO_COURSE_LIST*.

Preview the query and compare the output to the activity results.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Select the PSU_COURSE_TBL record.
3. Select the COURSE and DESCR fields.
4. Select *RFT* long in the DESCR field properties.
5. Save the query as NO_COURSE_LIST.

Adding a Subquery

To add a subquery:

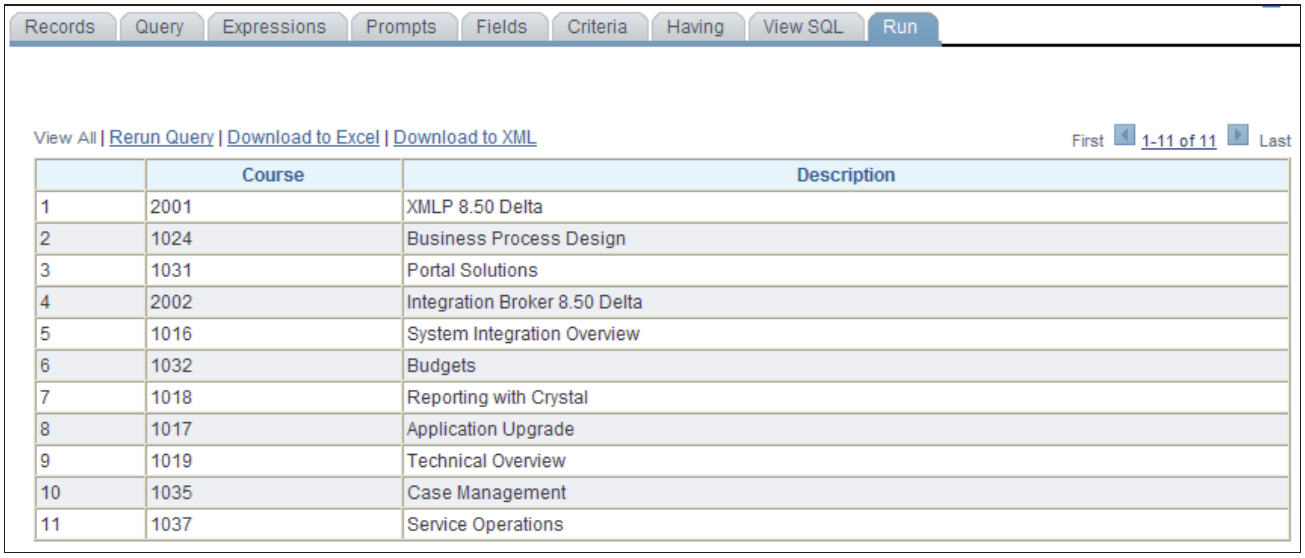
1. Select the Criteria tab, and click the Add Criteria button.
2. Enter the following information on the Edit Criteria Properties page:

Page Element	***Value or Status***
Expression 1 Type	*Field*
Expression 1	*COURSE*
Condition Type	*not in list*
Expression 2 Type	*Subquery*

3. Click the Define/Edit Subquery link.
4. Search and add the PSU_STU_ENROLL record.
5. Click the Select link for the COURSE field.
6. Save and preview the query.
7. Compare the output with the following results.

Results

The NO_COURSE_LIST query returns 11 rows:



	Course	Description
1	2001	XMLP 8.50 Delta
2	1024	Business Process Design
3	1031	Portal Solutions
4	2002	Integration Broker 8.50 Delta
5	1016	System Integration Overview
6	1032	Budgets
7	1018	Reporting with Crystal
8	1017	Application Upgrade
9	1019	Technical Overview
10	1035	Case Management
11	1037	Service Operations

This concludes the activity. Please do not continue.

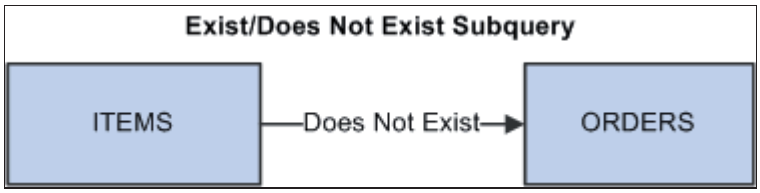
Creating an Exists or Does-Not-Exist Subquery

Using Exists or Does-Not-Exist Subqueries

Exists or *does-not-exist* subqueries are also known as true/false subqueries.

The condition that is selected *(exists* or *does-not-exist)* determines the data returned.

This diagram shows an example of using the *exist* or *does-not-exist* subquery:



Slide 179

Student Notes

Exists or Does-Not-Exist Subqueries

In the previous example diagram, items are returned in the primary query, and orders are used in the subquery. When the condition *does-not-exist* is selected, items appear only if no related entries exist in the Orders table.

The *exists* and *does-not-exist* conditions do not find matching items in a list; rather, they search for related rows, one row at a time. You do not select a field in the subquery; rather, you use criteria to check for the *existence* of related rows.

You create *exists* (true) or *does-not-exist* (false) subqueries to check for the existence or nonexistence of a condition and to return information to the primary query.

Does-Not-Exist Subquery SQL

Here is an example SELECT statement:

```
SELECT  A.ITEM_CD,
         A.DESCR
FROM  PS_PSU_ITEM_TBL A
WHERE  DOES NOT EXIST
         (SELECT 'X' FROM PS_ORD_DTL B WHERE B.ITEM_CD = A.ITEM_CD)
```

An exists or does-not-exist subquery requires a row of criteria in order for rows to be returned.

In this example, you see this criterion:

```
B.ITEM_CD = A.ITEM_CD
```

This table explains the *does-not-exist* subquery SQL:

SQL	***Explanation***
`SELECT A.ITEM_CD, A.DESCR` `FROM PS_PSU_ITEM_TBL A`	Retrieve the item code and description columns from the item table.
`WHERE DOES NOT EXIST`	Retrieve only the values that do not exist in the result set of the subquery.

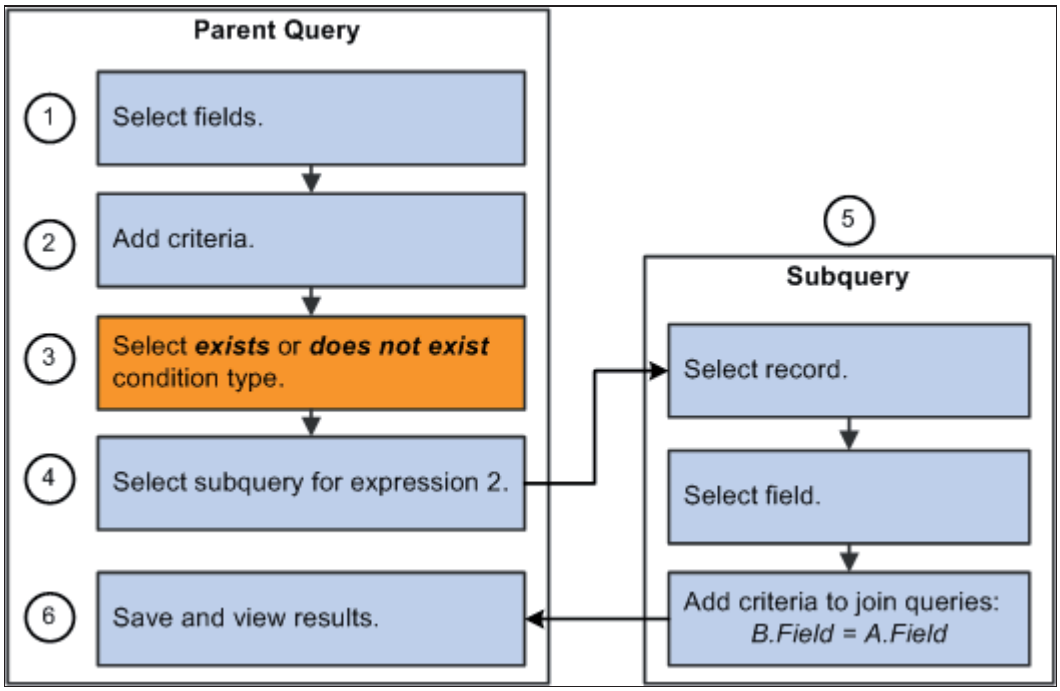
`(SELECT 'X'`
`FROM PS_ORD_DTL`
`WHERE B.ITEM_CD = A.ITEM_CD)`
	Retrieve only the items that do not appear on a purchase order detail line in the order detail table.

Note. When the retrieved data of the subquery depends on data retrieved in the primary query, the subquery must contain a WHERE clause to link itself to the primary query. The subquery cannot run independently.

Creating an Exists or Does-Not-Exist Subquery (continued)

Steps for Creating an Exists or Does-Not-Exist Subquery

Use the following steps to create an exists or does-not-exist subquery:



Slide 180

Student Notes

Adding Criteria – Step 2

The row of criteria used to create or attach the subquery has nothing on the Expression 1 side of the equation. After you select a field for Expression 1, *Exists or Does Not Exist*is no longer a valid option.

Also, more than one row of criteria might link the primary and subquery (that is, multiple common keys).

Activity 26: Creating a Does-Not-Exist (True/False) Subquery

In this activity, you will review the activity overview and create a *does-not-exist* subquery.

Slide 181

Activity Overview

Create the NO_ITEM_ORDERS query for the Inventory department.

The query displays the item codes and descriptions of only items that do not have orders associated with them. The item table (PSU_ITEM_TBL) stores item data. The order detail table (ORD_DTL) store order line items.

Preview the query and compare the output to the results of the activity.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Does-Not-Exist Subquery

To create a does-not-exist subquery:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Use the following specifications to begin the query:

Record	***Fields***
PSU_ITEM_TBL	ITEM_CD DESCR, *RFT Long*

3. Save the query as NO_ITEM_ORDERS.
4. Select the Criteria page, and then click the Add Criteria button.
5. Enter the following information on the Edit Criteria Properties page:

Page Element	***Value or Status***
Expression 1 Type	*Field*
Expression 1	*ITEM_CD*
Condition Type	*does not exist*
Expression 2 Type	*Subquery*

6. Click the Define/Edit Subquery link.
7. Search for and add the ORD_DTL record.
8. Select the Criteria page, and click the Add Criteria button.

9. Enter this information on the Edit Criteria Properties page of the subquery:

Page Element	***Value or Status***
Expression 1 Type	*Field*
Expression 1	*B.ITEM_CD*
Condition Type	*equal to*
Expression 2 Type	*Field*
Expression 2	*A.ITEM_CD*

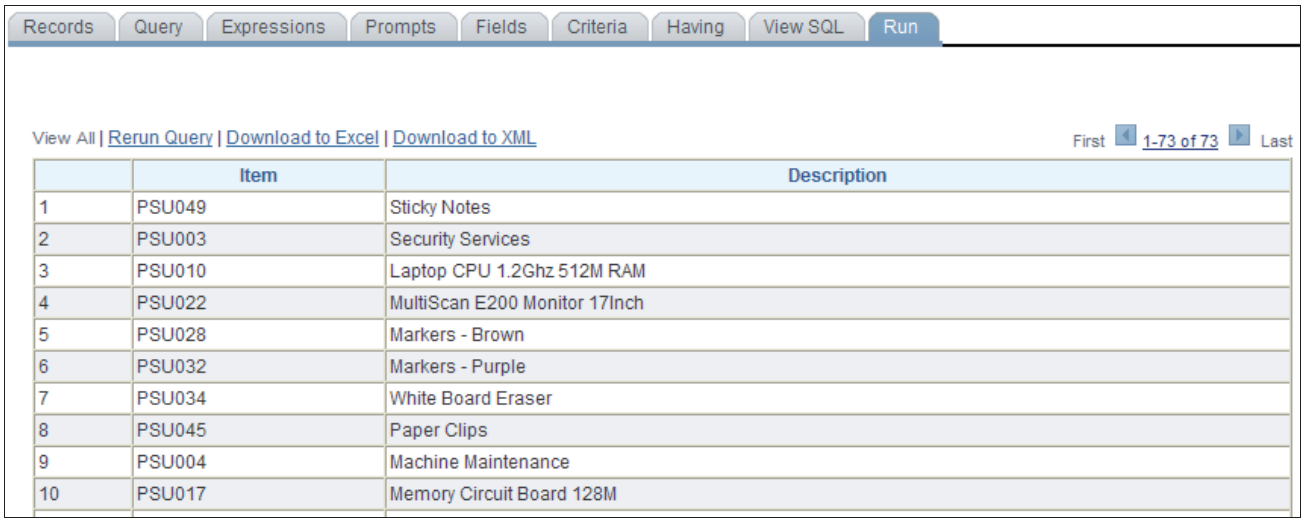
10. Click the OK button.

11. Save and preview the query.

12. Compare the output with the following results.

Results

The NO_ITEM_ORDERS query returns 73 rows:



	Item	Description
1	PSU049	Sticky Notes
2	PSU003	Security Services
3	PSU010	Laptop CPU 1.2Ghz 512M RAM
4	PSU022	MultiScan E200 Monitor 17Inch
5	PSU028	Markers - Brown
6	PSU032	Markers - Purple
7	PSU034	White Board Eraser
8	PSU045	Paper Clips
9	PSU004	Machine Maintenance
10	PSU017	Memory Circuit Board 128M

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- A subquery is a query within a query, and you use subqueries to compare a value for a field in the primary query to the results of a subordinate query.
- You can use a single-value subquery to check for the existence or nonexistence of a condition or to return information to the primary query.
- You can select the In List or Not in List condition type to efficiently retrieve data from the subquery.
- Creating an exists or does-not-exist subquery requires you to link the subquery to the primary query; you accomplish this by creating a row of criteria in the subquery.

Slide 182

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Explaining subqueries Creating a single-value subquery	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Advanced Query Options"

Lesson 14

Working with Unions

Objectives

By the end of this lesson, you will be able to:

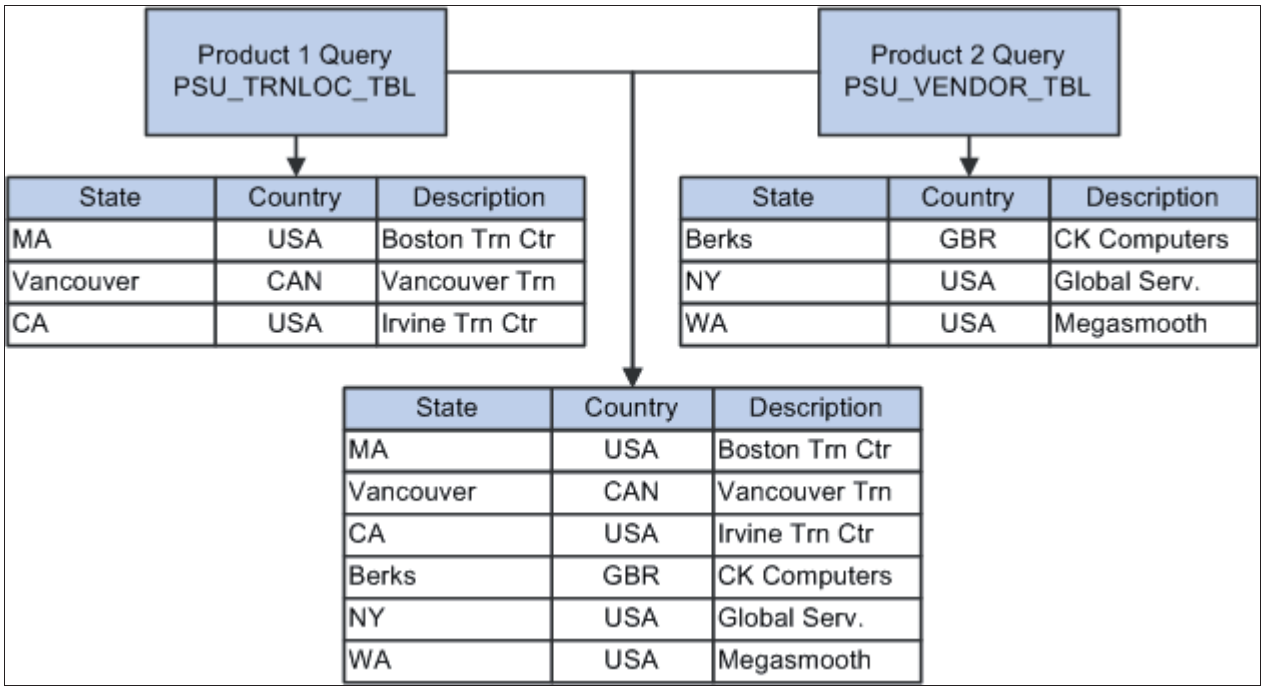
- Explain unions.
- Use literals as placeholder fields.
- Create unions.
- View union SQL.

Slide 184

Explaining Unions

Two SELECT Statements Used

You use unions to combine records that have no fields in common and to retrieve similar data from unrelated records in one query, as shown in this example:



Slide 185

Student Notes

Union Rules

A union is two SELECT statements that you combine in the same query.

Unions enable you to have two tables in the same query without having joining criteria and without creating a Cartesian product.

You must follow three rules with unions:

- Each statement must consist of the same number of fields.
- Each statement must consist of the same data types.
- The field data types in each statement should correspond (be in the same order).

Note. You do *not* have to use fields of like data type in one statement. That is, you can mix field data types within each statement as long as the data types correspond between the two statements.

Example: Two SELECT Statements Used

Suppose that the Training department requires a query that lists all training locations and all vendors from which it buys products. To obtain a consolidated listing of all locations and vendors, you need to combine the names of the training locations from the PSU Training Locations table (PSU_TRNLOC_TBL) with the names of the vendors from the Vendor table (PSU_VENDOR_TBL.)

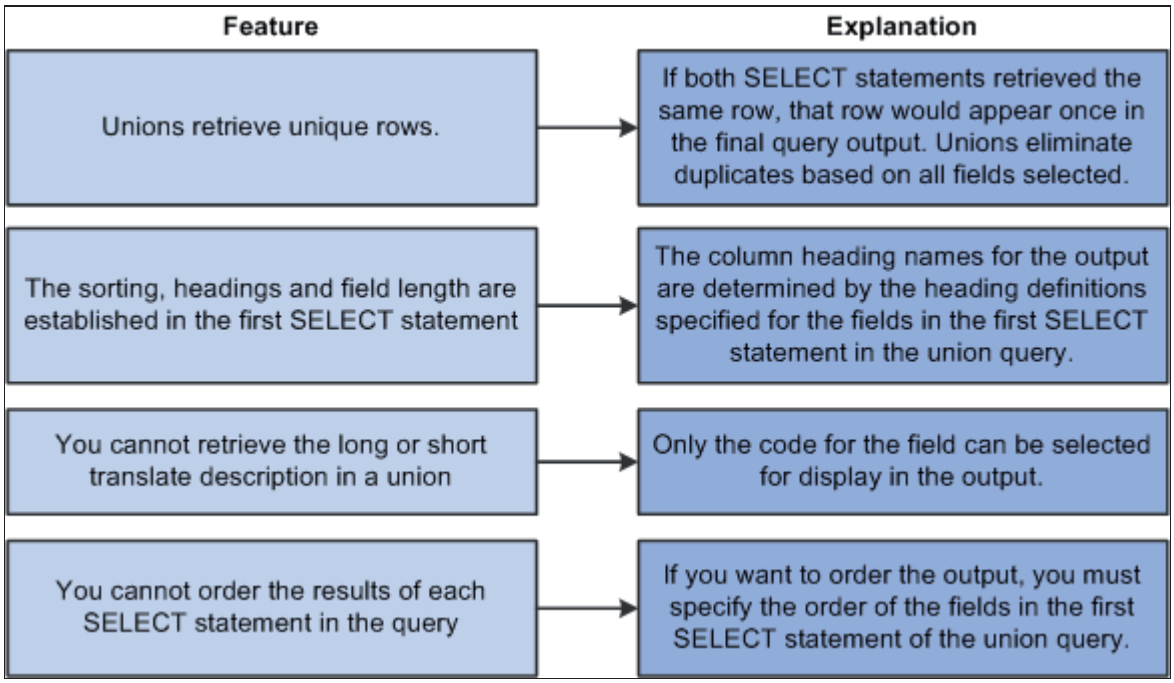
To retrieve the correct query output, you would write two separate statements and create a union to combine them in one query, as shown in this example:

```
SELECT  A.DESCR     CHAR30
        A.STATE     CHAR6
        A.COUNTRY   CHAR3
FROM PS_PSU_TRNLOC_TBL A
UNION
SELECT  B.DESCR     CHAR30
        B.STATE     CHAR6
        B.COUNTRY   CHAR3
FROM    PS_PSU_VENDOR_TBL B
```

Explaining Unions (continued)

Union Features

This diagram lists and explains the features of union:



Slide 186

Student Notes

Platform Requirements

On some platforms, if matching fields have different lengths, the larger one must be in the first SELECT statement. Other platforms require that the matching fields have the same length.

Union SQL

This is an example of Union SQL:

```
SELECT   A.TRAINING_LOC     CHAR6
         A.DESCR            CHAR30
         A.COUNTRY          CHAR3
         A.DESCRSHORT       CHAR10
FROM     PS_PSU_TRNLOC_TBL  A
UNION
SELECT   B.VENDOR           CHAR6
         B.DESCR            CHAR30
         B.COUNTRY          CHAR3
         B.DESCRSHORT       CHAR10
FROM     PS_PSU_VENDOR_TBL  B
```

Using Literals as Placeholder Fields

Using Literals in Unions

Literals are placeholders, or pieces of text. Literals are useful when you create complex unions.

When you create a union, both queries must have the same number of fields, but the fields do not have to be identical.

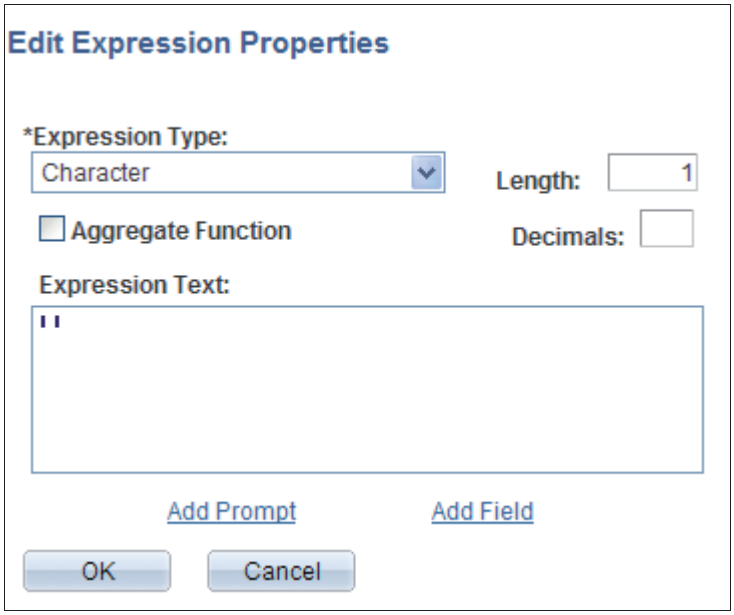
Therefore, applying literals as placeholder fields is useful.

Slide 187

Student Notes

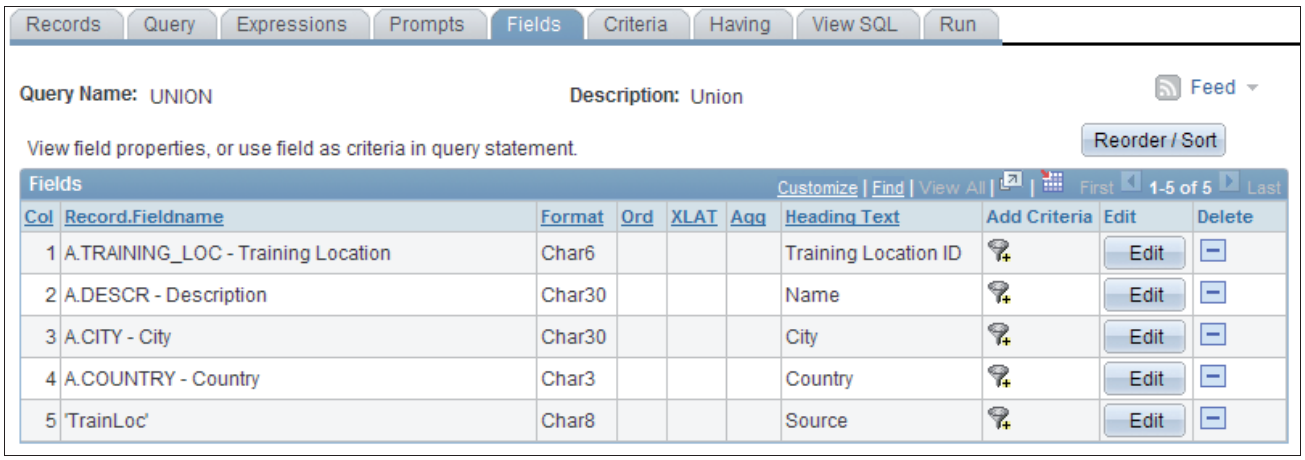
Literals Used as Placeholders

Use a literal expression as a placeholder when creating unions, as shown here:



Unions require that you use the same numbers of fields in each SELECT statement. Because the fields do not have to be the same, you can create a literal expression and use the expression as a field.

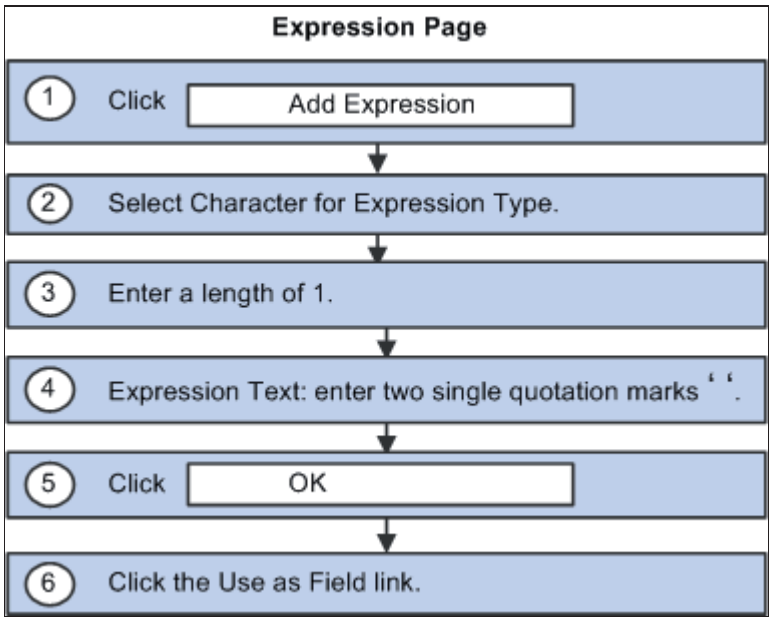
This example shows five fields, and the fifth ('TrnLoc') is a literal field:



Using Literals as Placeholder Fields (continued)

Creating a Literal

Perform these steps to use literals as placeholder fields:



Slide 188

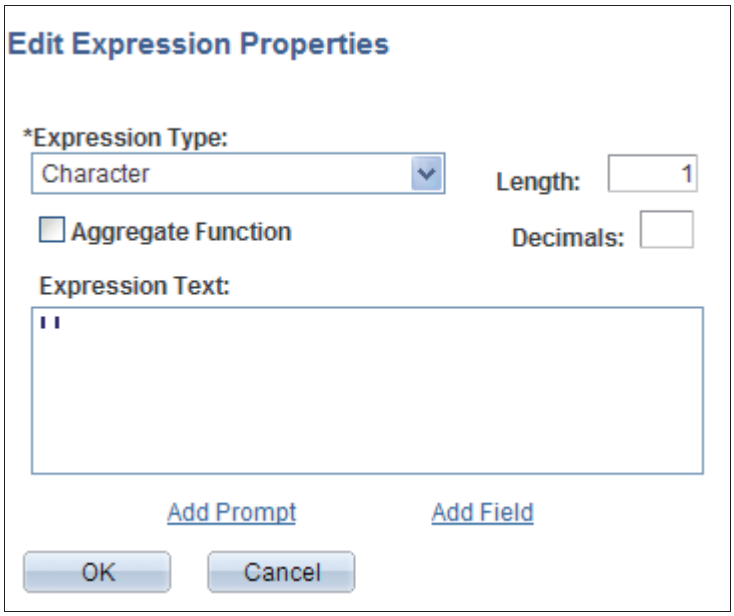
Student Notes

Steps for Using Literals in Unions

To use literals in unions:

1. Click the Add Expression button on the Expressions page:

2. Select *Character* as the expression type, enter *1* as the length, enter two single quotes (' ') as the expression text, and then click the OK button:



Note. Enclose the literal value between the single quotes. Change the length according the text requirement.

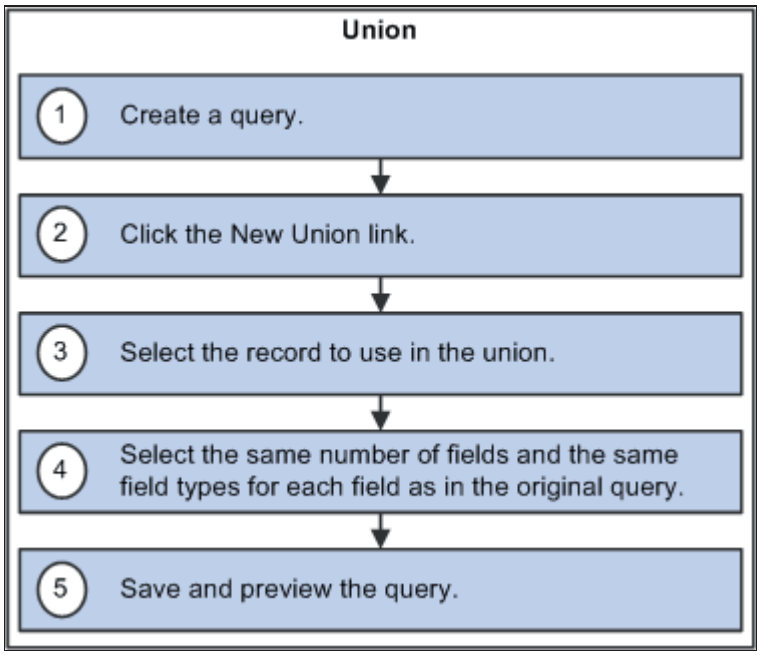
3. Click the OK button to return to the Expression page.

4. Click the Use as Field link to use the expression in the query:

Creating Unions

Creating Unions

This diagram shows the steps used to create a union:



Slide 189

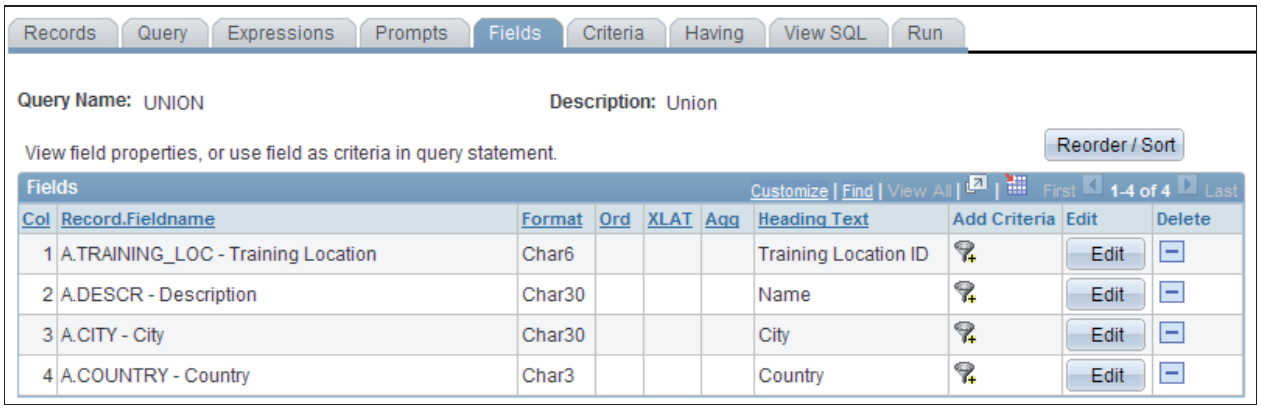
Student Notes

Example: Creating Unions That Use Literals

Follow these steps to create unions:

1. Create the first query.

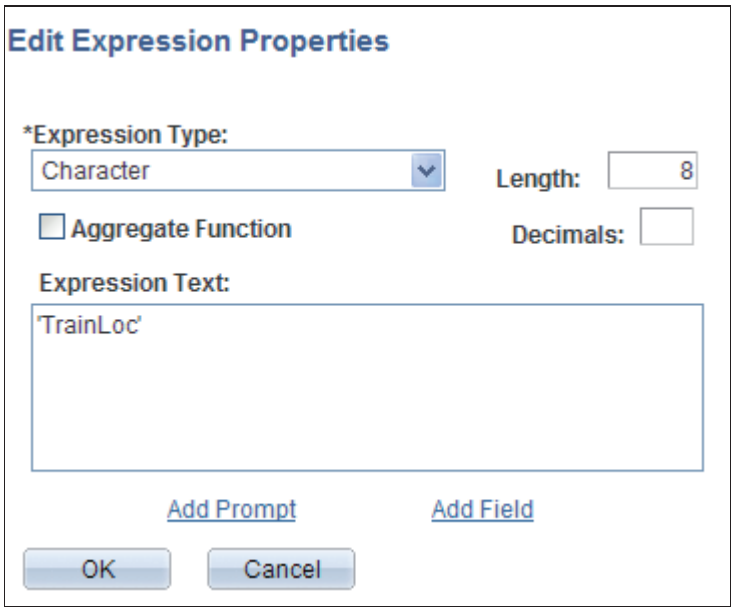
   For example, create the UNION query using the PSU_TRAINING_LOC record.



2. Access the Expression page, and click the Add Expression button.

3. In the Edit Expression Properties page, set the literal field.

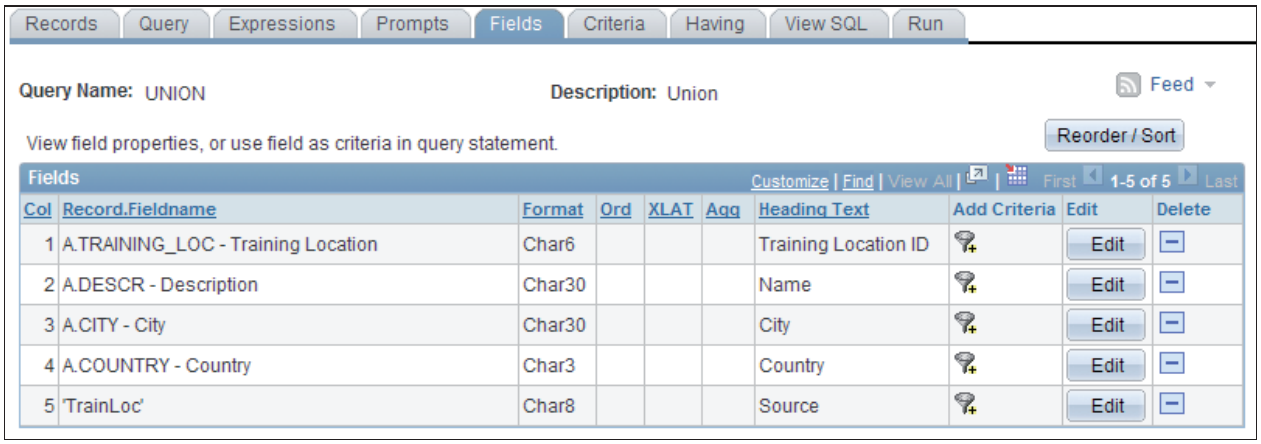
   In this example, the literal is 'TrainLoc'.



4. Click the OK button to return to the Expression page, and click the Use as Field link.

5. Access the Fields page, and modify the heading text as needed.

   This example shows the Fields page that include a literal field, 'TrainLoc':



   The literal field called 'TrnLoc' populates the literal column with the word *TrnLoc* if the row comes from the training location record.

Important! PeopleSoft Query orders the columns and sorts the rows based on what you specify in the Top Level of Query selection. It also uses the column headings that you define for the first selection. Changing these properties in the second query does nothing.

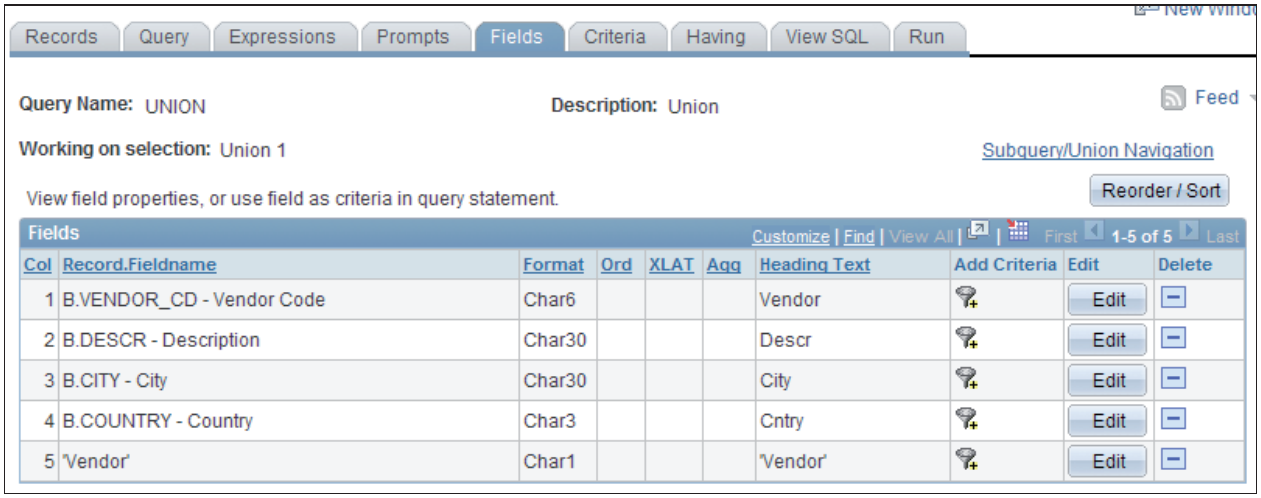
---

6. Click the New Union link to add the second record and fields.

   In this example, add the PSU_VENDOR_TBL record and its fields.

7. Create the second literal field.

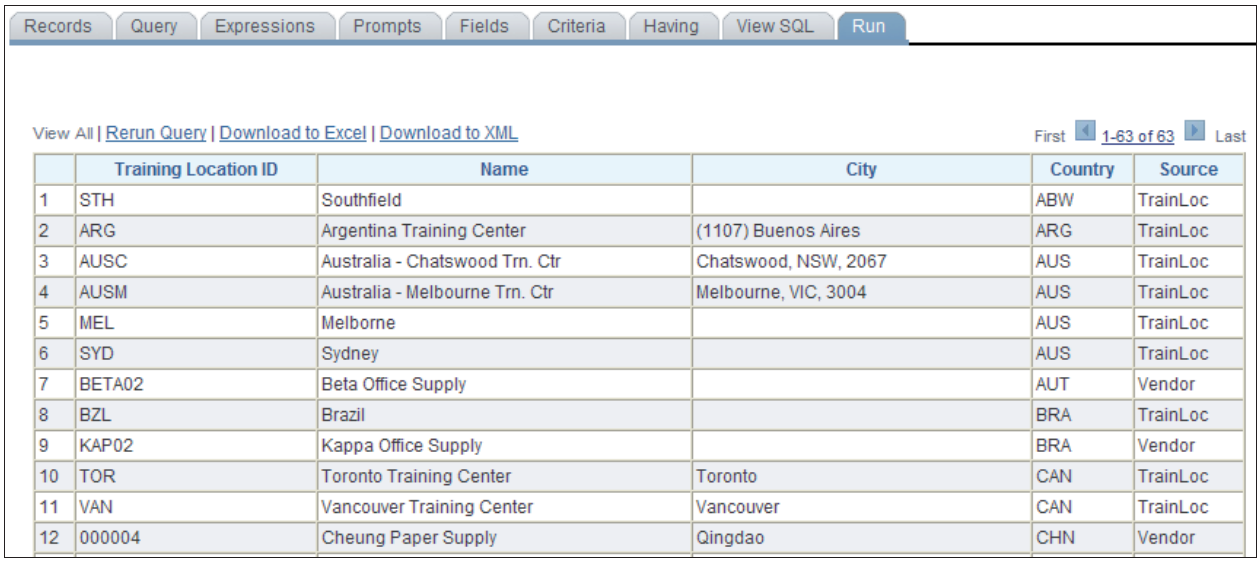
   In this example, create the literal field called 'Vendor', and notice the fields and the field properties in this query:



   The literal field called 'Vendor' populates the literal column with the word *Vendor* if the row comes from the vendor record.

8. Save and run the query.

   In this example, notice the Country and Source columns; the query results are displayed by country and indicate which company is a training location and which is a vendor:



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

View All | Rerun Query | Download to Excel | Download to XML — First 1-63 of 63 Last

	Training Location ID	Name	City	Country	Source
1	STH	Southfield		ABW	TrainLoc
2	ARG	Argentina Training Center	(1107) Buenos Aires	ARG	TrainLoc
3	AUSC	Australia - Chatswood Trn. Ctr	Chatswood, NSW, 2067	AUS	TrainLoc
4	AUSM	Australia - Melbourne Trn. Ctr	Melbourne, VIC, 3004	AUS	TrainLoc
5	MEL	Melborne		AUS	TrainLoc
6	SYD	Sydney		AUS	TrainLoc
7	BETA02	Beta Office Supply		AUT	Vendor
8	BZL	Brazil		BRA	TrainLoc
9	KAP02	Kappa Office Supply		BRA	Vendor
10	TOR	Toronto Training Center	Toronto	CAN	TrainLoc
11	VAN	Vancouver Training Center	Vancouver	CAN	TrainLoc
12	000004	Cheung Paper Supply	Qingdao	CHN	Vendor

Generated SQL

The union query in this example determines which companies are training locations and which ones are vendors.

The first SELECT statement of this query would look like this:

```
SELECT A.COUNTRY       CHAR3
       A.TRAINING_LOC  CHAR6
       A.DESCR         CHAR3
       A.DESCRSHORT    CHAR10
       'TrnLoc'        CHAR8
FROM   PS_PSU_TRNLOC_TBL A
```

The second SELECT statement of this query would look like this:

```
UNION
SELECT  B.COUNTRY     CHAR3
        B.VENDOR      CHAR6
        B.DESCR       CHAR3
        B.DESCRSHORT  CHAR10
        'Vendor'      CHAR8
FROM PS_PSU_VENDOR_TBL B
ORDER BY COUNTRY
```

Viewing Union SQL

Identify Union SQL Issues

To troubleshoot a query if an error occurs at runtime, you can view the SQL that the union generates. Possible SQL problems to check are:

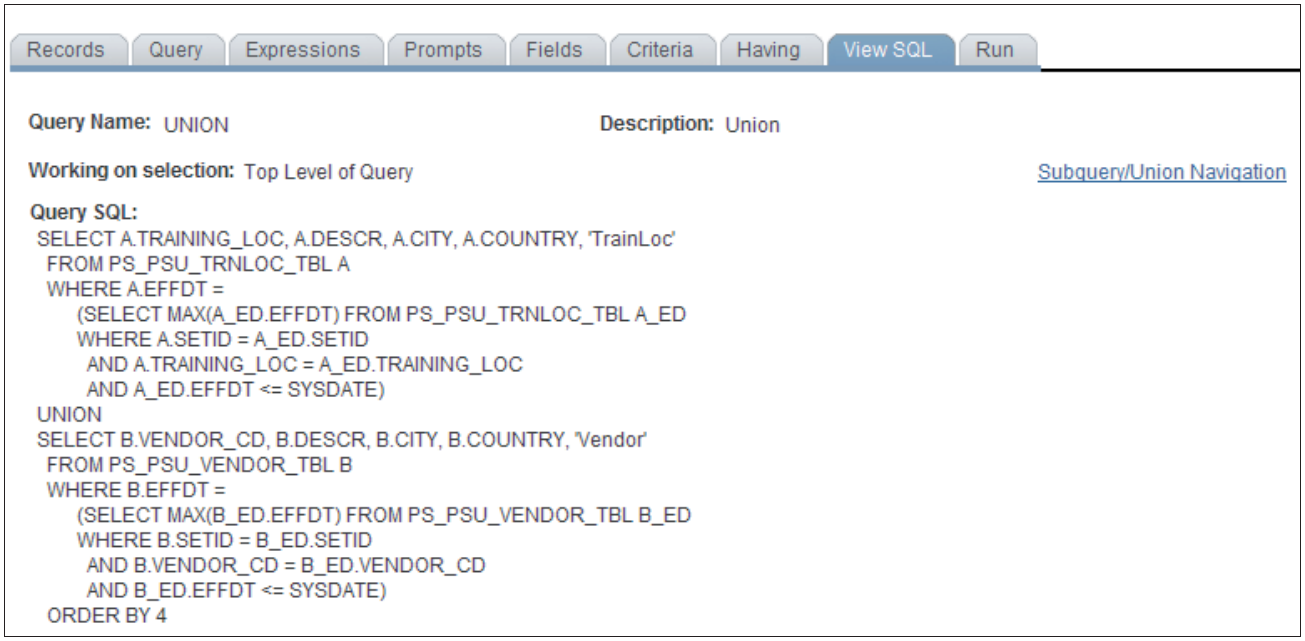
- ☑ Both Select statements do not have the same number of fields.
- ☑ Fields are out of order.
- ☑ A union has been established.

Slide 190

Student Notes

Union SQL Explained

To view the SQL, select the View SQL page for either query. You locate existing problems by viewing the SQL that the query generates, as shown here:



Activity 27: Creating Queries with Unions

In this activity, you will review the activity overview and:

1. Create a query.
2. Create a literal expression.
3. Create a union.

Slide 191

Activity Overview

Marketing needs you again. This time, the department wants a combined list of training locations and vendors. The department needs the ID codes, names, cities, countries, and a field that identifies the line as a training location or vendor. Display the information based on country. The training location (PSU_TRNLOC_TBL) and the vendor (PSU_VENDOR_TBL) records store this information.

Use UNION as the query name.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Access Query Manager, and create a new query using the following information:

Record	*Fields*
PSU_TRNLOC_TBL	TRAINING_LOC DESCR COUNTRY CITY

Note. Click the OK button to dismiss the effective-dated record message dialog box.

3. Save the query as UNION.

Creating a Literal Expression

To create the literal expression:

1. Select the Expression tab, and then click the Add Expression button.
2. Enter the following information:

Page Element	*Value or Status*
Expression Type	*Character*
Length	*8*
Expression Text	*'TrnLoc'*

3. Click the OK button, and click the Use as Field link.

4. Edit the field properties using the following information:

Field	***Property***
TRAINING_LOC	*Text* *ID*
DESCR	*Text* *Name*
COUNTRY	*RFT Long*
'TrnLoc'	*Text* *Source*

5. Sort the query using the country field, and save the query.

Creating a Union

To create a union:

1. Click the New Union link in Query Manager.
2. Search for the PSU_VENDOR_TBL record, and then click the Add Record link.
3. Click the OK button to accept the effective-dated record.
4. Select the following fields:

Page Element	***Value or Status***
VENDOR_CD	Selected
DESCR	Selected
CITY	Selected
COUNTRY	Selected

5. Select the Expressions page, and then click the Add Expression button to create a literal expression for vendor.

6. Enter the following information:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*8*
Expression Text	*'Vendor'*

7. Click the OK button, and click the Use as Field link.
8. Save and preview the query.
9. Compare your results to the following results.

Results

The UNION query returns 63 rows:

	ID	Name	City	Country	Source
1	STH	Southfield		ABW	TrnLoc
2	ARG	Argentina Training Center	(1107) Buenos Aires	ARG	TrnLoc
3	AUSC	Australia - Chatswood Trn. Ctr	Chatswood, NSW, 2067	AUS	TrnLoc
4	AUSM	Australia - Melbourne Trn. Ctr	Melbourne, VIC, 3004	AUS	TrnLoc
5	MEL	Melborne		AUS	TrnLoc
6	SYD	Sydney		AUS	TrnLoc
7	BETA02	Beta Office Supply		AUT	Vendor
8	BZL	Brazil		BRA	TrnLoc
9	KAP02	Kappa Office Supply		BRA	Vendor
10	TOR	Toronto Training Center	Toronto	CAN	TrnLoc

This concludes the activity. Please do not continue.

Activity 28: Using Joins in Unions

In this activity, you will review the activity overview and:

1. Create a query.
2. Create literal expressions.
3. Create a union.
4. Create a join.

Slide 192

Activity Overview

Create the NO_DUPS query. Display the course description, course type, and course code from the Course record (PSU_COURSE_TBL) and display session location and instructor from the Session record (PSU_CRS_SESSN). The data that you retrieve should not be duplicated for each session, and session information should not be displayed on the same row as the data that is retrieved from PSU Course record.

You need to create the union using the PSU Course Session record (PSU_CRS_SESSN) and join that table to the PSU Training Locations record. Create literals for the union.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

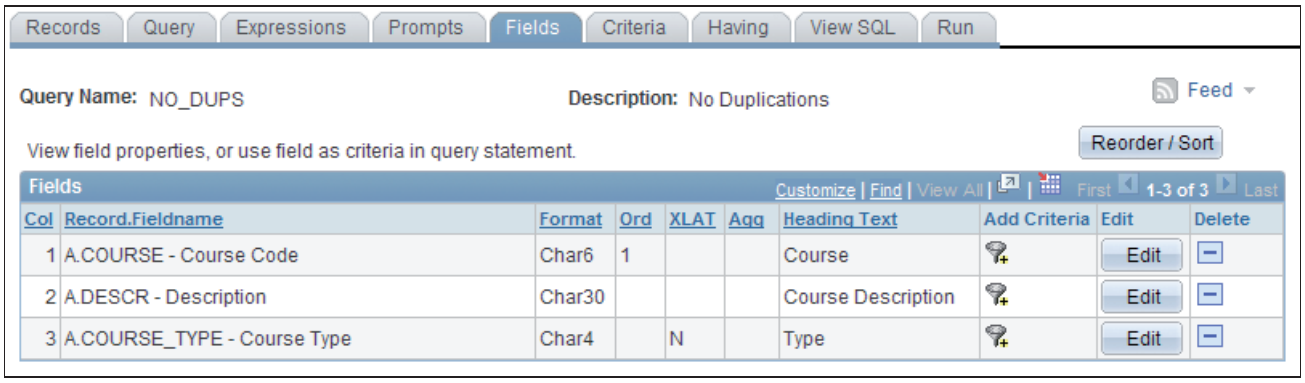
1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Use the PSU_COURSE_TBL record with these fields and settings:

Field	***Properties***
COURSE	*Order by 1*
DESCR	*Text* *Course Description*
COURSE_TYPE	*RFT Short*

3. Save the query as NO_DUPS.

Results

These are the results of creating the query:



Creating Literal Expressions

To create literal expressions:

1. Select the Expression tab, and click the Add Expression button.

2. Create the first literal expression for the parent query using the following information:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*3*
Expression Text	*' '* (two single quotes)

3. Click the OK button, and click the Use as Field link.
4. Access the Expressions page, and click the Add Expression button.
5. Create the second literal expression for the parent query using the following information:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*30*
Expression Text	*' '* (two single quotes)

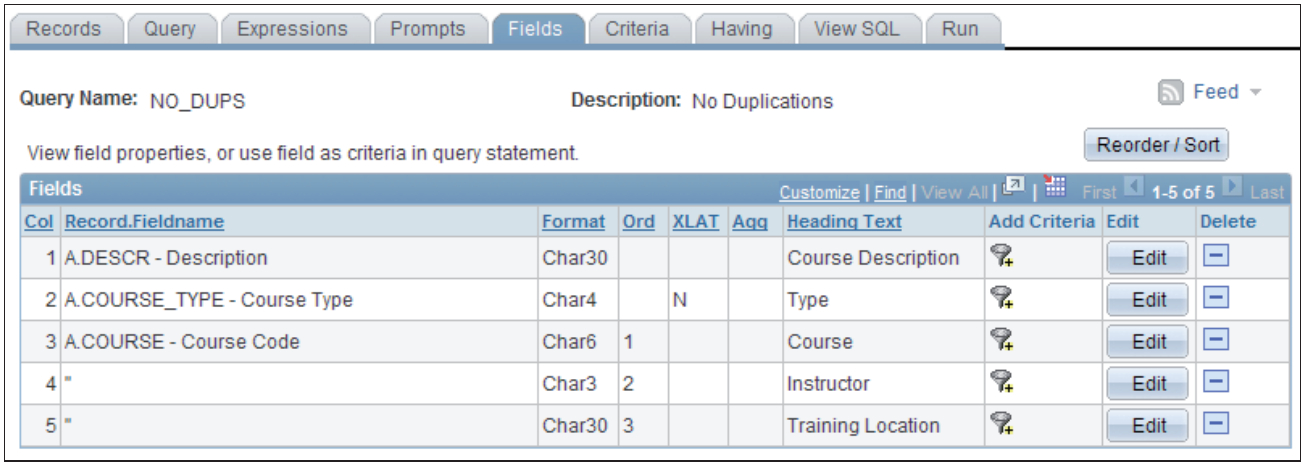
6. Click the OK button, and click the Use as Field link for the second literal expression.
7. In the Fields page, click the Edit button for the literal with three characters, enter *Instructor* as the heading text, and click the OK button.
8. Click the Edit button for the literal with 30 characters, and enter *Training Location* as the heading text, and click the OK button.
9. Click the Reorder/Sort button, and enter the following information:

Field	***New Column***	***New Order By***
A.COURSE	*3*	*1*
A.DESCR	*1*	<Blank>
A.COURSE_TYPE	*2*	<Blank>
' ' (Instructor)	<blank>	*2*
" (Training Location)	<blank>	*3*

10. Click the OK button, and save the query.

Results

This example shows the Fields page after you create the literal expressions:



Creating a Union

To create a union:

1. In Query Manager, click the New Union link.
2. Search for and add the PSU_CRS_SESSN record to create the union.
3. Select the following fields:

Field	***Value or Status***
COURSE	Selected
INSTRUCTOR	Selected

4. Save the query.

Creating a Join

To create a join:

1. On the Query page, click the Join PSU_TRNLOC_TBL-PSU Training Locations related-record link.
2. Accept the standard join, and click the OK button.
3. Click the OK button to accept the effective date criteria.
4. Select the DESCRSHORT field.
5. Select the Expressions tab, and then click the Add Expression button.

6. Enter the following information for the first literal expression of the second query:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*30*
Expression Text	*' '* (two single quotes)

7. Click the OK button, and click the Use as Field link.
8. Access the Expressions page, and click the Add Expression button.
9. Enter the following information for the second literal expression of the second query

Page Element	***Value or Status***
Expression Type	*Character*
Length	*3*
Expression Text	*''* (two single quotes)

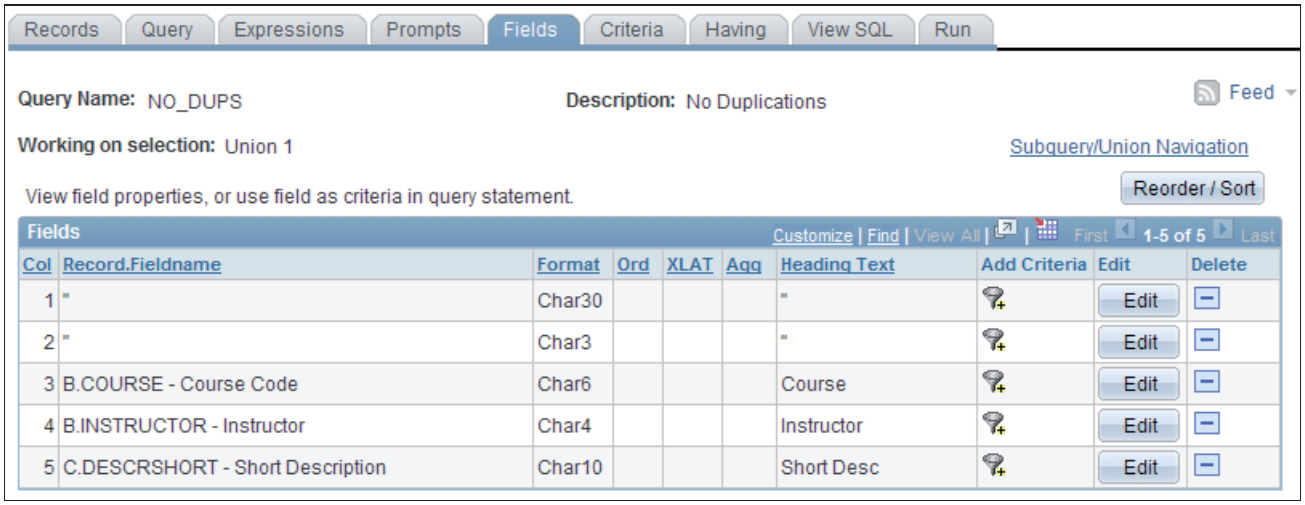
10. Click the OK button, and click the Use as Field link for the second expression.
11. Click the Reorder/Sort button, and enter the following information:

Page Element	***New Column***
B.COURSE	*3*
B.INSTRUCTOR	*4*
C.DESCRSHORT	*5*
Expression 1 (Char30)	*1*
Expression 2 (Char3)	*2*

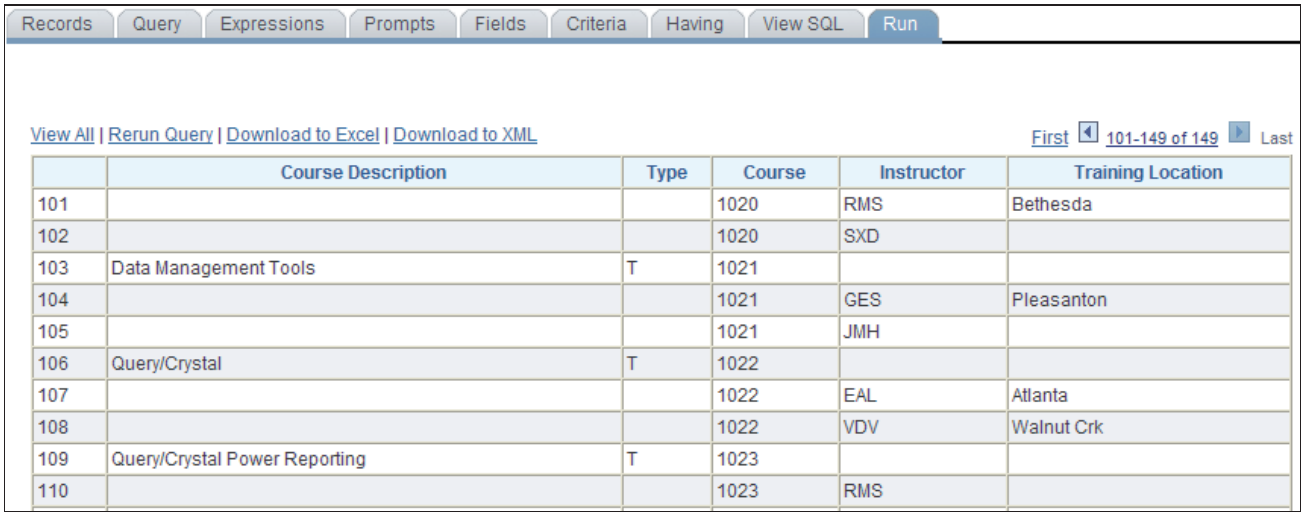
12. Save the query, and run the query.
13. Compare the output with the following results.

Results

This example show the Fields page after you complete creating join:



The NO_DUPS query returns 149 rows:



This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- Unions combine the result sets of two queries into one result set that limits duplicates by default.
- Literals are useful in unions as placeholders and as identifiers.
- You click the New Union link in Query Manager to create unions.
- You can view the SQL code for a union by selecting the View SQL page.

Slide 193

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Explaining unions	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Advanced Query Options"

Lesson 15

Performing Outer Joins

Objectives

By the end of this lesson, you will be able to:

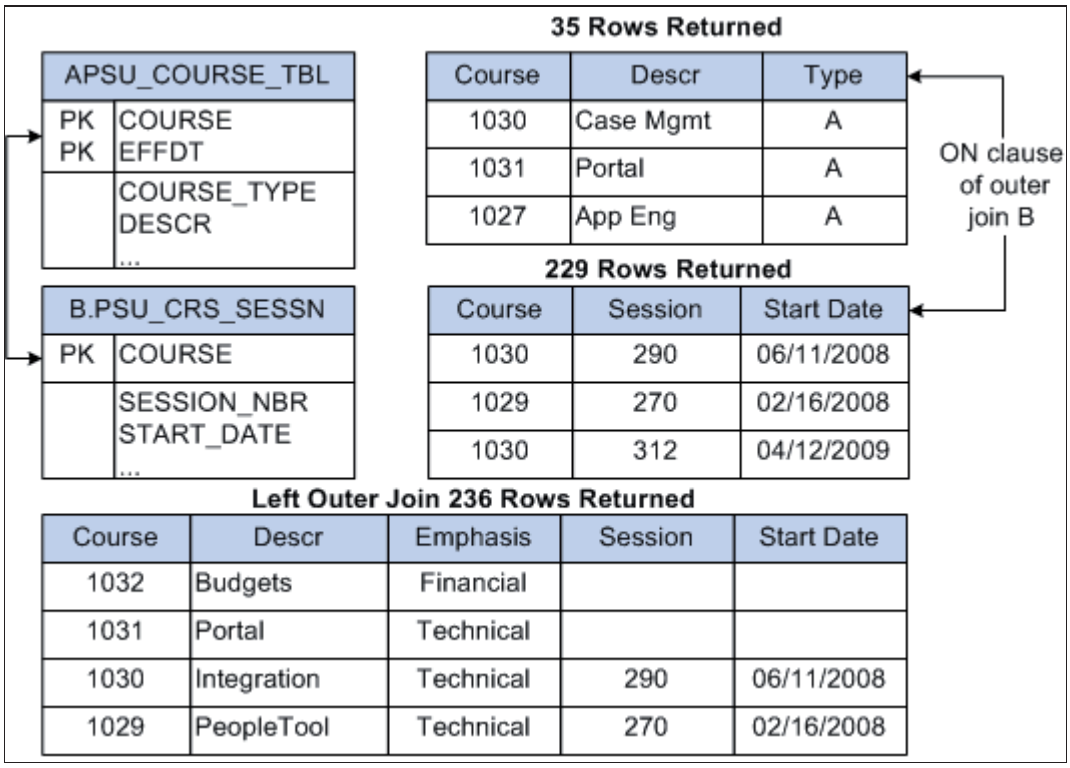
- Describe outer joins.
- Create standard outer joins.

Slide 195

Describing Outer Joins

Outer Joins

This example shows an outer join:



Slide 196

Student Notes

Purpose of Outer Joins

In an outer join, all rows of data are included from the master table. Matching rows from the subordinate table are also included.

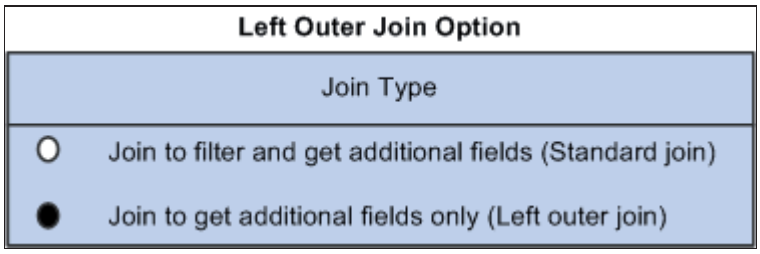
With a left outer join, all rows from table 1 [for example, the Course table (PSU_COURSE_TBL)] appear in the result, even if no match is in table 2 {for example, no course IDs are in the Course Session table (PSU_CRS_SESSN)].

Outer joins combine aspects of record-hierarchy joins and false subqueries. Remember that a record-hierarchy join retrieves rows for which fields match from different tables, for example, A.Field1 equals B.Field1. A false subquery retrieves rows that do not exist in a secondary table.

Creating Standard Outer Joins

Outer Join Option

You create an outer join when you select a different join type, as shown in this example:

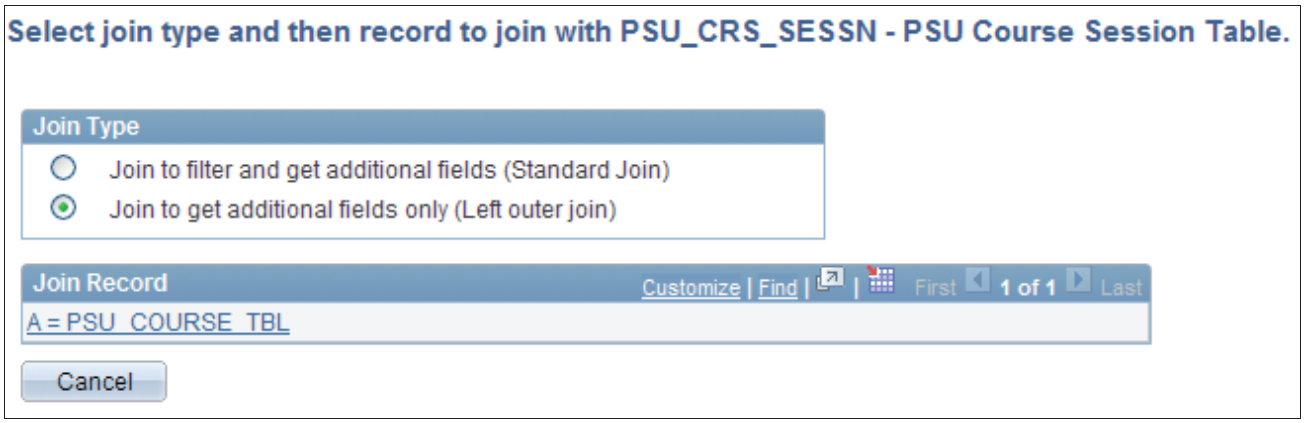


Slide 197

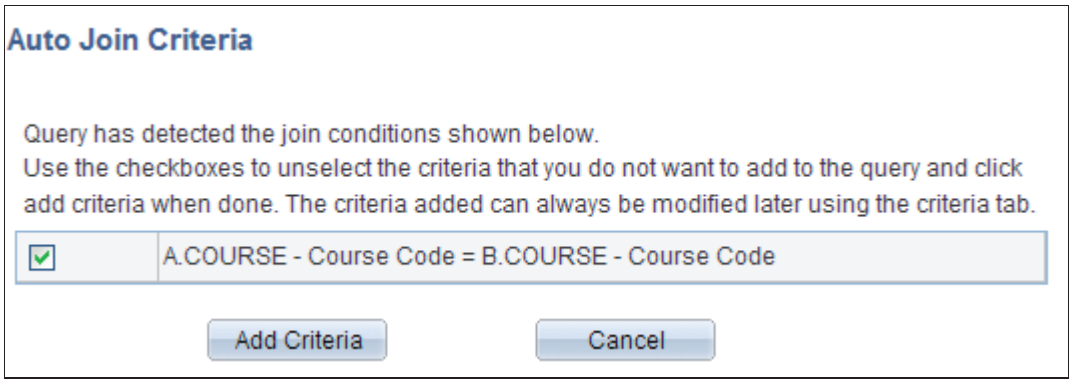
Student Notes

Select Join Type

When you join records, the Select Join Type page appears so that you can select the type of join for the query:



After you click the join record link, the Auto-join Criteria page appears, enabling you to add a row of criteria to the query:

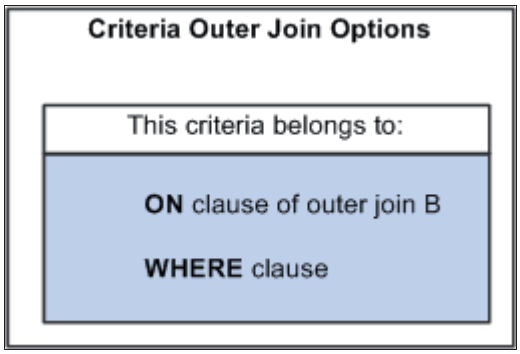


Creating Standard Outer Joins (continued)

Outer Join Options on the Criteria Page

Because Query Manager offers left outer join capabilities as of PeopleTools release 8.44, a new drop-down list appears on the Edit Criteria Properties page.

This example shows the two options on the Criteria page:

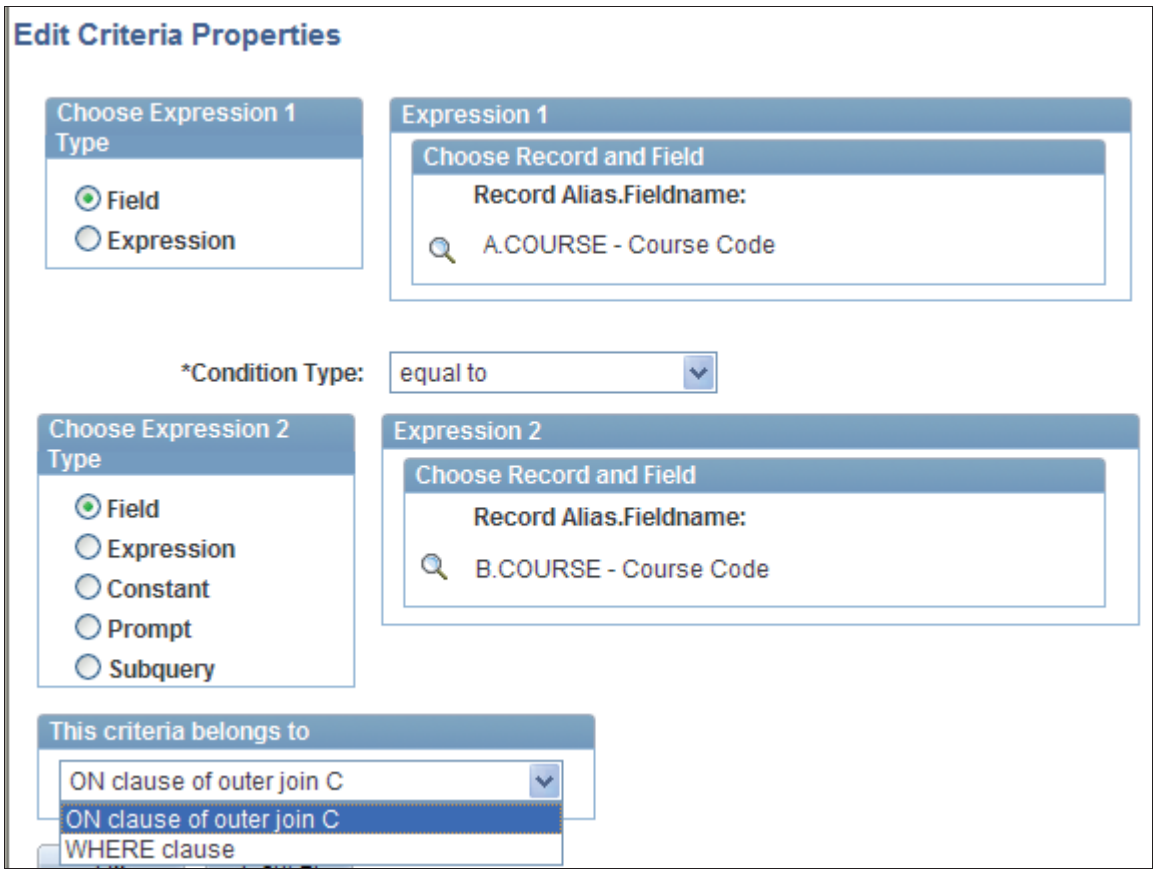


Slide 198

Student Notes

Criteria Outer Join Options

The Edit Criteria Properties page provides a drop-down list in which you can select whether the criteria belongs to the *On clause for outer joins* or the*WHERE clause*for other join types:



Criteria Placement

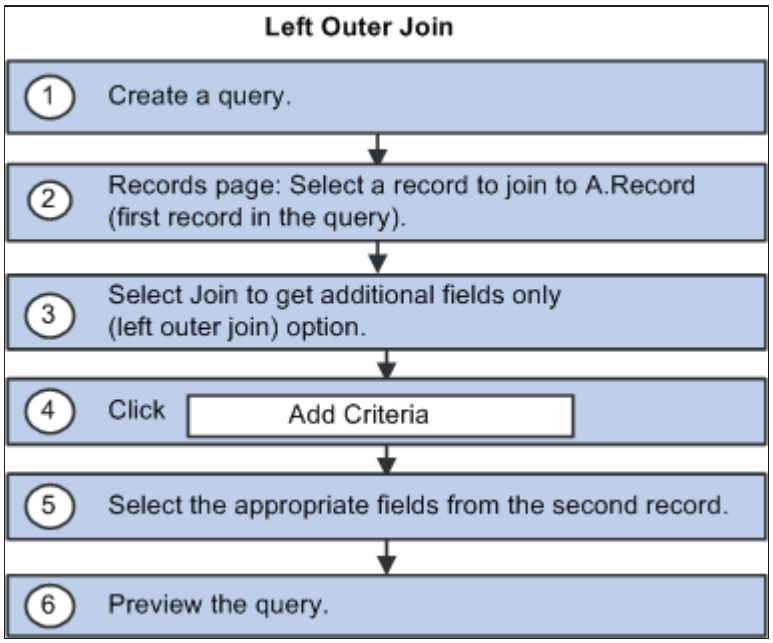
Place criteria on the WHERE clause when the criteria applies to the primary query.

Place criteria on the ON clause when the criteria apply to the subordinate query only.

Creating Standard Outer Joins (continued)

Creating a Left Outer Join

Perform these steps to create a left outer join:



Slide 199

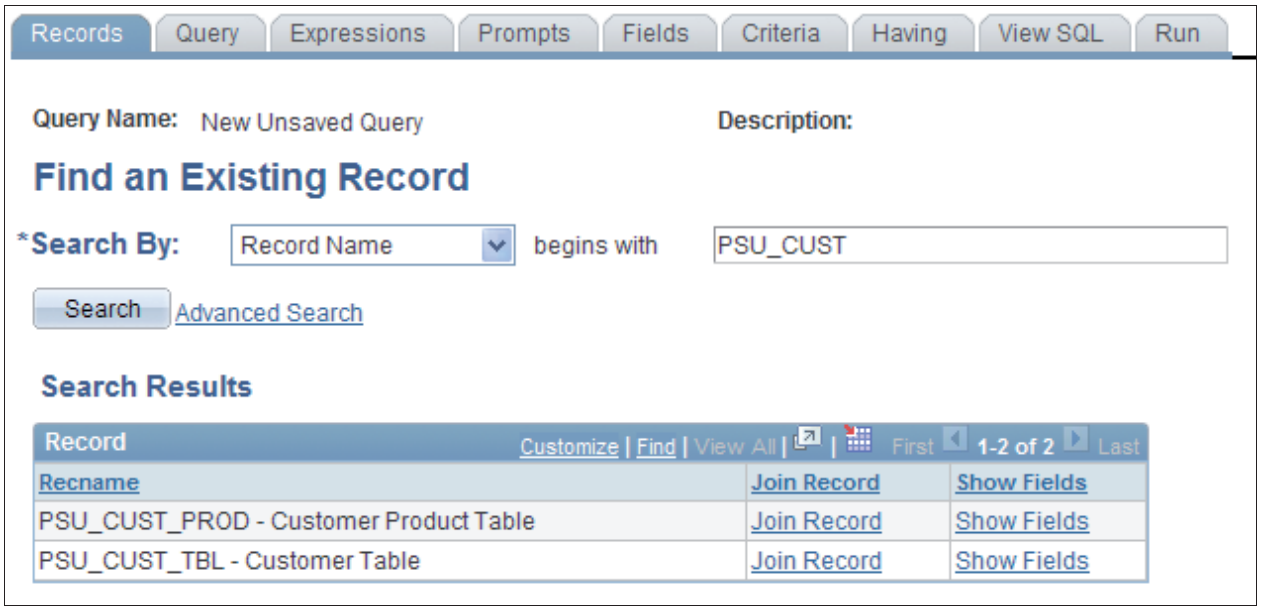
Student Notes

Outer Join Steps

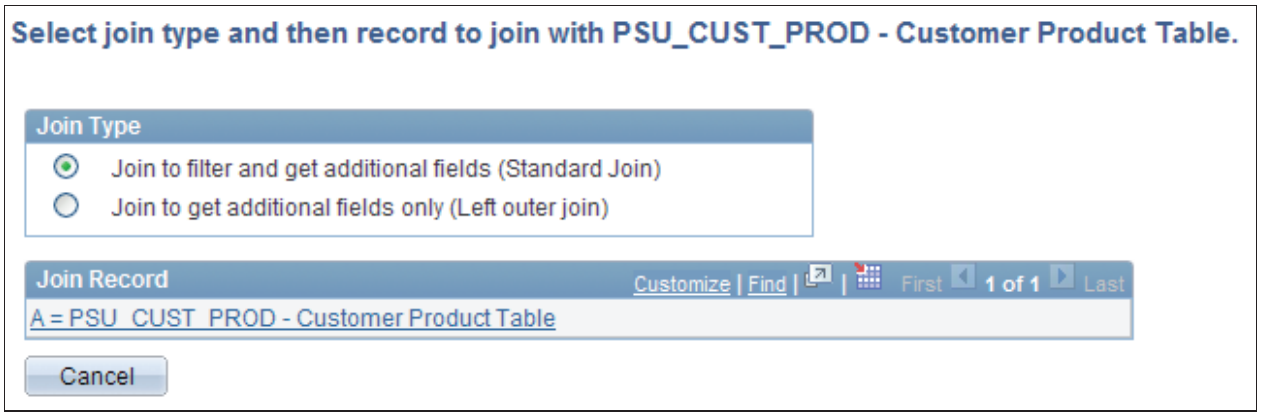
Use the following pages and steps to create a left outer join:

1. Create a query using at least one record and one field.

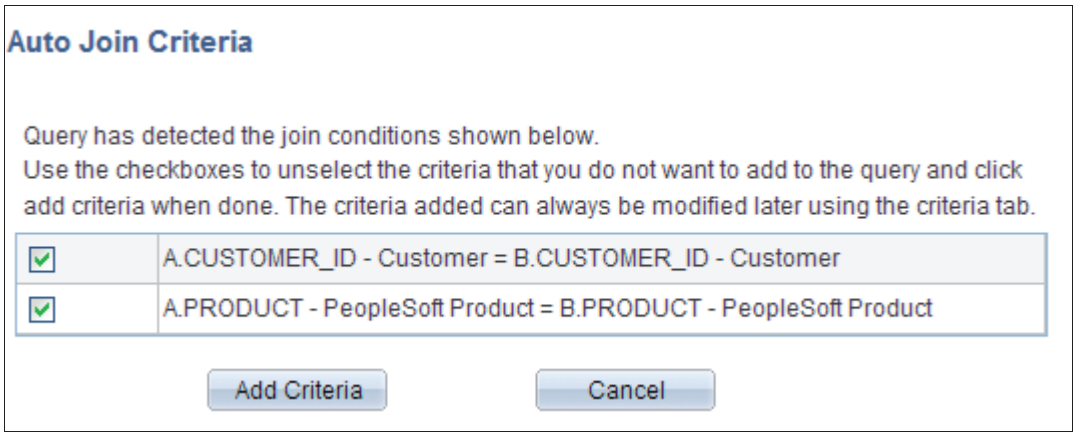
2. Select the Records page, and click the Join Record link to begin creating the outer join:



3. Select the second join type option (left outer join), and click the join record link:



4. Click the Add Criteria button on the Auto Join Criteria page to insert the join criteria:



Activity 29: Creating Outer Joins

In this activity, you will review the activity overview and:

1. Create a query.
2. Create an outer join.

Slide 200

Activity Overview

Create the SESSIONS_INQUIRY query.

Display the course code, description, course type, session number, and session date for all scheduled and unscheduled classes. Order the display by date and then by course.

Retrieve the data from the Course (PSU_COURSE_TBL) and the Course Sessions (PSU_CRS_SESSN) records.

Note. Use *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Query

To create a query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Access Query Manager, and create a new query using the PSU_COURSE_TBL record and the following settings:

Field	***Value or Status***
COURSE	Selected *RFT Long*
DESCR	Selected *Text* *Course Name*
COURSE_TYPE	Selected

3. Save the query as SESSIONS_INQUIRY.

Creating an Outer Join

To create an outer join:

1. Select the Record page, and search for the PSU_CRS_SESSN record.
2. Select the *Join to get additional fields only (left outer join)* option, and click theA = PSU_COURSE_TBL join record link.
3. Click the Add Criteria button.
4. Select the SESSION_NBR and START_DATE fields.
5. Select the Fields page, and select *RFT Long* as the START_DATE field heading.
6. Click the Reorder/Sort button, and enter this information:

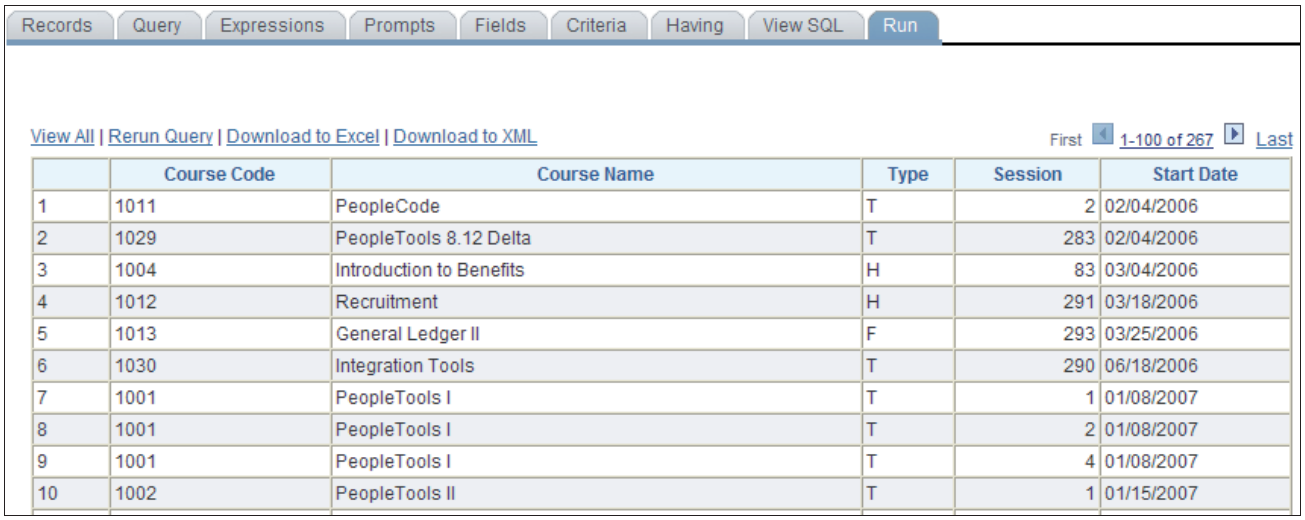
Page Element	***New Order By***
START_DATE	*1*
COURSE	*2*

7. Save and preview the query.

8. Compare the output with the following results.

Results

The SESSIONS_INQUIRY query returns 267 rows of results:



	Course Code	Course Name	Type	Session	Start Date
1	1011	PeopleCode	T	2	02/04/2006
2	1029	PeopleTools 8.12 Delta	T	283	02/04/2006
3	1004	Introduction to Benefits	H	83	03/04/2006
4	1012	Recruitment	H	291	03/18/2006
5	1013	General Ledger II	F	293	03/25/2006
6	1030	Integration Tools	T	290	06/18/2006
7	1001	PeopleTools I	T	1	01/08/2007
8	1001	PeopleTools I	T	2	01/08/2007
9	1001	PeopleTools I	T	4	01/08/2007
10	1002	PeopleTools II	T	1	01/15/2007

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- A left outer join retrieves all rows from table 1, even if no match exists for the key field in table 2.
- You create outer joins by selecting the option *Join to get additional fields only (Left outer join).*

Slide 201

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Creating Standard Outer Joins Describing outer joins	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,*"Advanced Query Options"

Lesson 16

Using Connected Query

Objectives

By the end of this lesson, you will be able to:

- Describe connected query.
- Use Connected Query Quick Start.
- Use Connected Query Manager.
- Use Connected Query Viewer.
- Use Connected Query Scheduler.

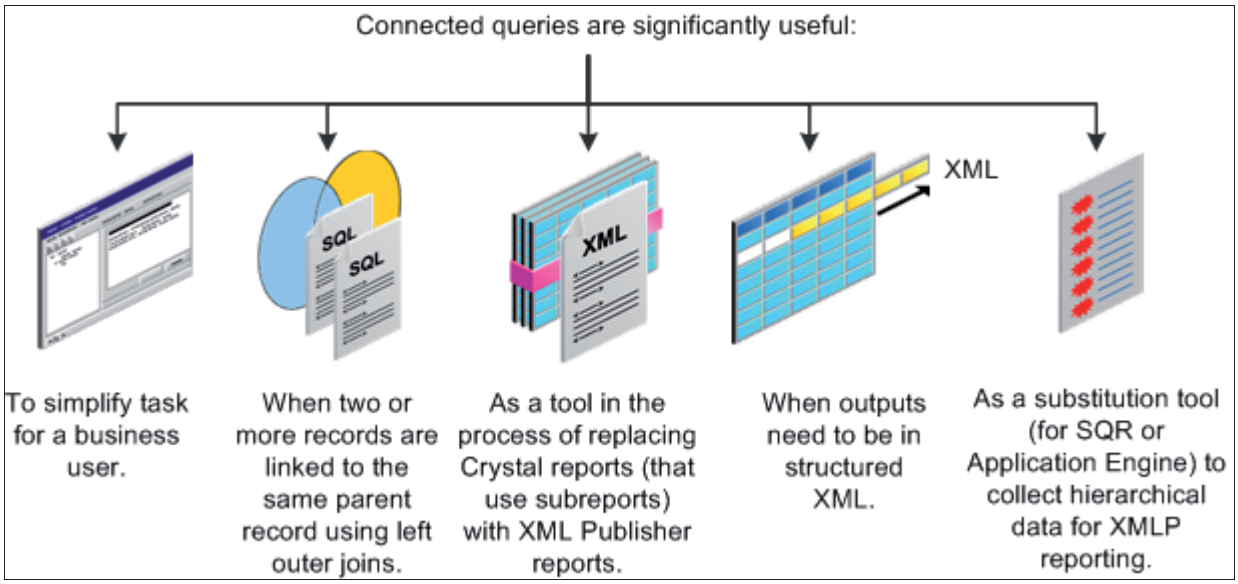
Slide 203

Describing Connected Query

Connected Query

Connected Query enables you to create a single XML file based on a set of queries with parent-child relationships.

This diagram shows the usages of connected queries:



Slide 204

Student Notes

Connected Query

A connected query is a hierarchical object built with existing PeopleSoft queries. A parent query can nest *n* levels of child queries and any child query can have *m* sibling queries within a hierarchy. There are no artificial limits for *n* and *m,* but typical use includes a single parent-child relationship or a few levels of nesting. Fields in a child query are mapped to related fields in the immediate parent query. A connected query returns a hierarchical data set in which data returned by child queries is filtered by the results of its immediate parent query.

Connected queries are used to analyze data, to supply PeopleSoft data to other systems using Web Services, and to create XML Publisher reports that use connected query as a Data Source. Connected queries are significantly useful:

- To simplify task for a business user.

  User can create a set of linked simple queries to collect data from multiple tables instead of creating a complicated query with multiple equal, outer joins, and grouping conditions.

- When two or more records are linked to the same parent record using left outer join.

  Connected Query enables you to collect data similar to query performing left outer joins. In contrast to query, Connected Query enables you to link multiple queries to a single parent query.

- As a tool in the process of replacing Crystal reports (that use subreports) with XML Publisher reports.

  Crystal reports can include the main reports and a set of subreports that could be combined in the same template. Using Connected Query, you can replace a Crystal reports having main reports and subreports with XML Publisher reports.

- When outputs need to be in structured XML.

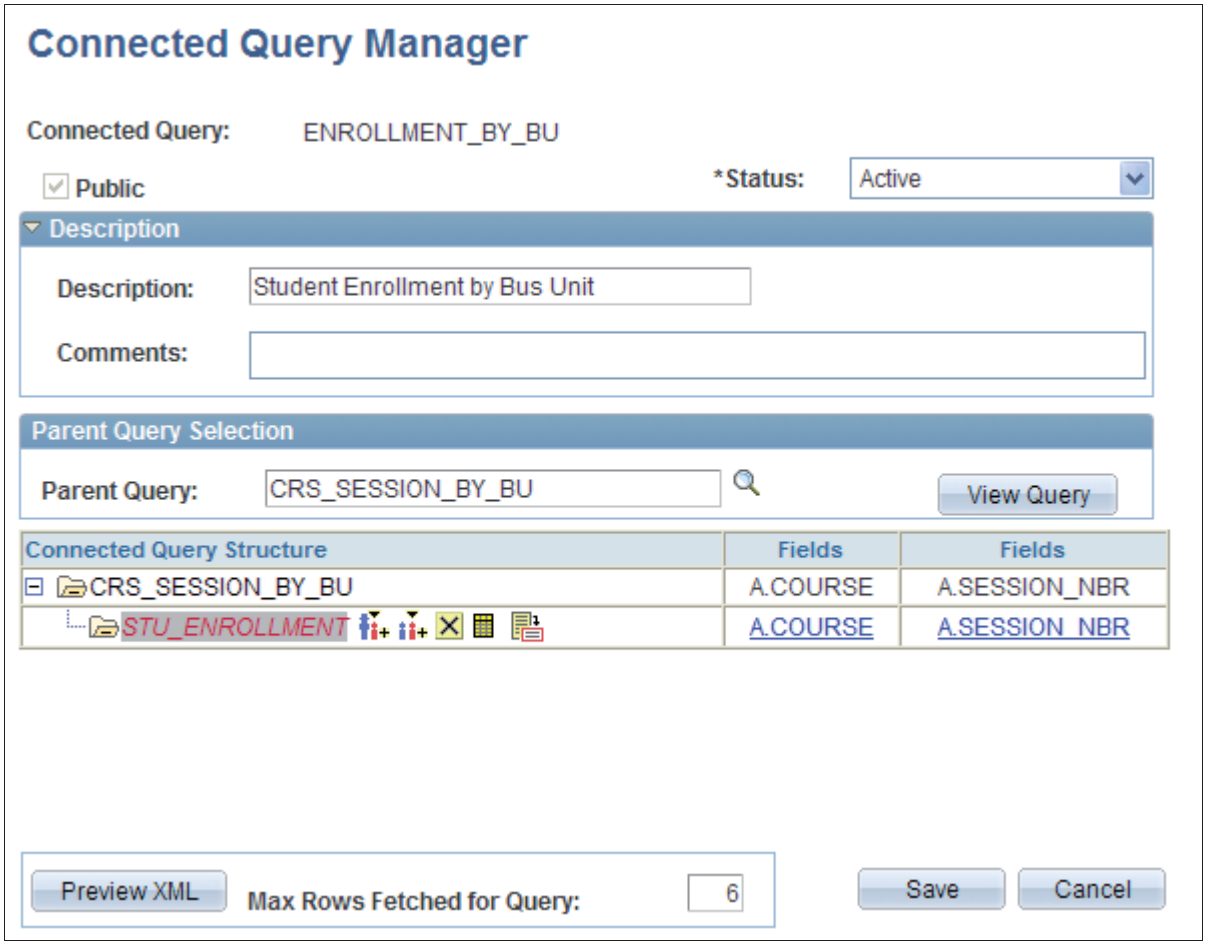
  A simple query (including a query with multiple joins) creates a tabular plain layout. To have a hierarchical output, you need to use multiple grouping conditions that involve complicated logic and is not always straight forward. Connected Query processing creates a hierarchical output where a single row of data from parent query results in a set of rows in a child query. This data is processed row by row and produces structured data. It has similarity with SQR and Application Engine nested loops processing. Instead of using Application Engine or SQR to collect data for file processing by XML Publisher, use Connected Query as the data collection tool because no conditional logic involves in this process. However, use SQR for processes with high volume of data or when high performance is required.

- As a substitution tool (for SQR or Application Engine) to collect hierarchical data for XMLP reporting.

Note. You can quickly create a single parent-child relationship using the Connected Query Quick Start component, and create complex connected queries using the Connected Query Manager.
Queries with in-tree prompt cannot be used in Connected Query.

Connected Query Example

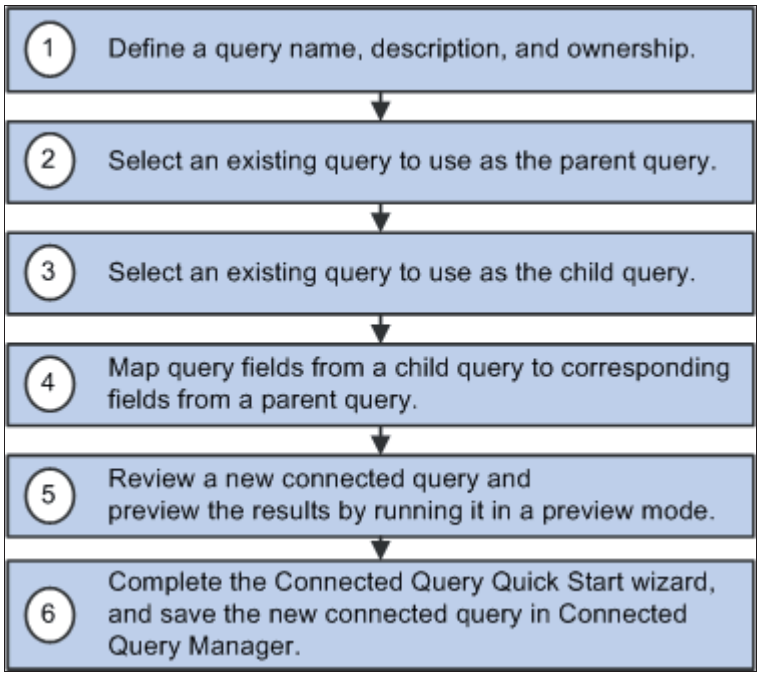
This example shows the structure that Connected Query uses to collect data of each course section that has students enrolled:



Using Connected Query Quick Start

Creating a Connected Query

This diagram lists a high-level overview of how to create a connected query using the Connected Query Quick Start wizard:



Slide 205

Student Notes

Connected Query Quick Start

The Connected Query Quick Start is a wizard that walks you through creating a simple connected query with a single parent query and a single child query linked using a set of related fields.

Note that:

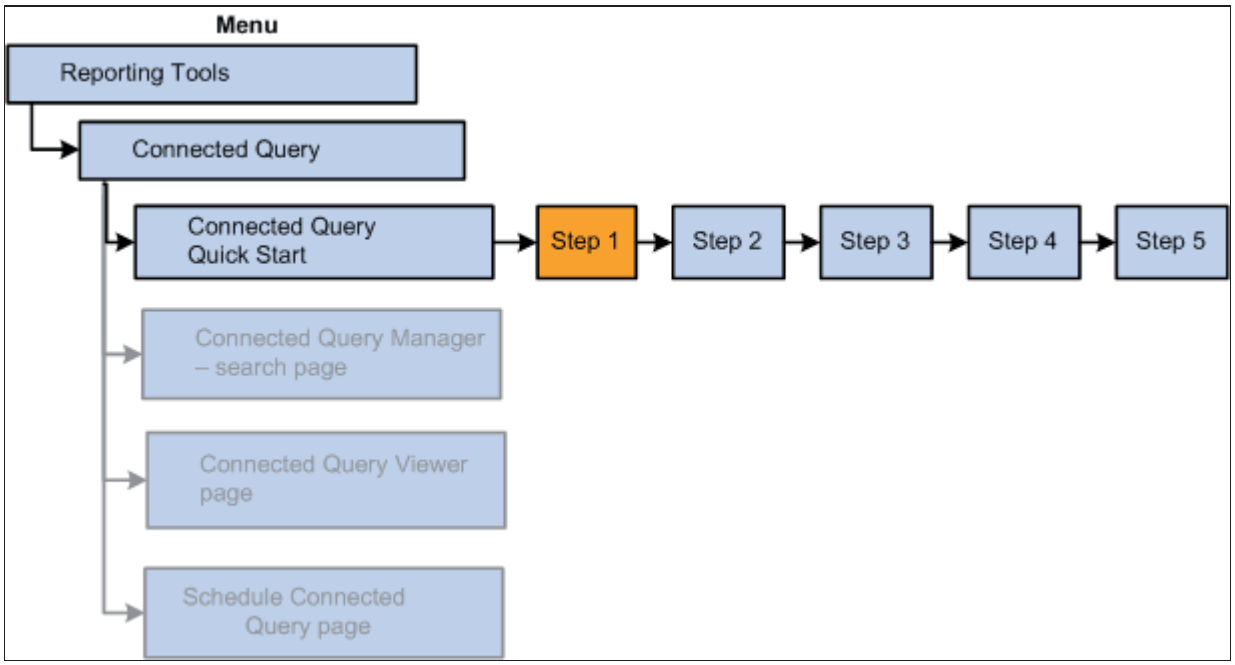
- The parent and child queries must exist and be visible in Query Manager before you can use them to create a connected query.
- Connected Query Quick Start is used only for creating new connected queries.

  You can edit existing connected queries using the Connected Query Manager page.
- Connected query uses query security permissions for each of its member queries.

Using Connected Query Quick Start (continued)

Connected Query Quick Start: Step 1

This diagram shows the navigation path to access the Connected Query Quick Start - Step 1:



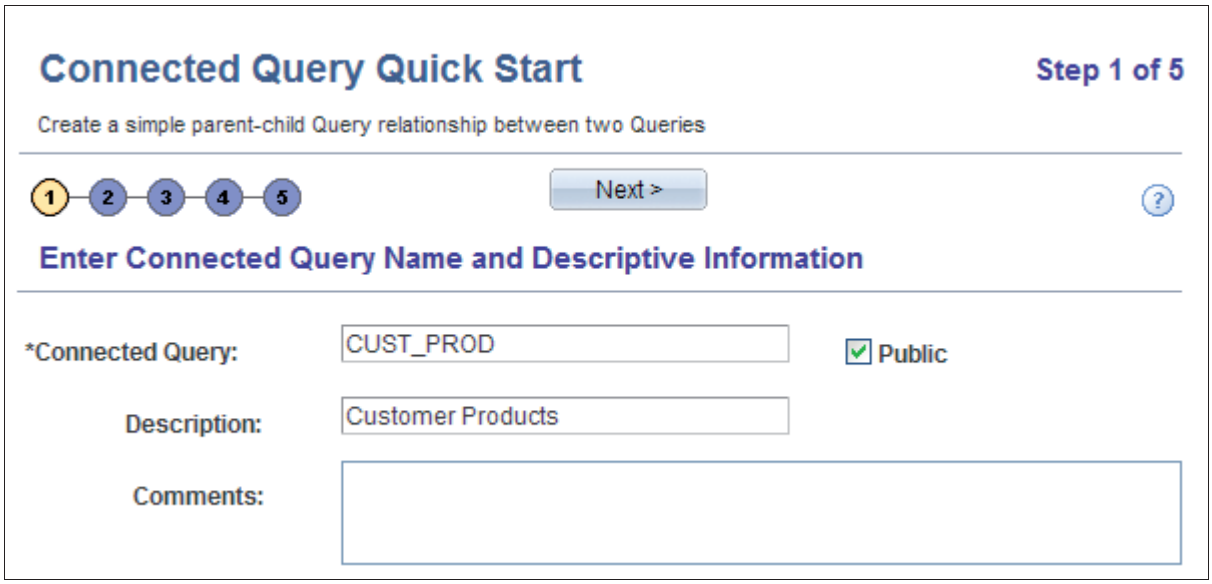
Slide 206

Student Notes

Page Used to Enter Descriptive Information of the Connected Query

Use this page to enter a connected query name and descriptive information, and to define the ownership of the connected query:

Page Name	*Navigation*
Step 1: Enter a Name and Descriptive Information	Reporting Tools, Connected Query, Connected Query Quick Start



Elements of the Enter a Name and Descriptive Information Page

Connected Query Enter a name for the connected query.

Next This button is available after you enter the name of the connected query and move to any other field.

Click to advance the wizard to the next page.

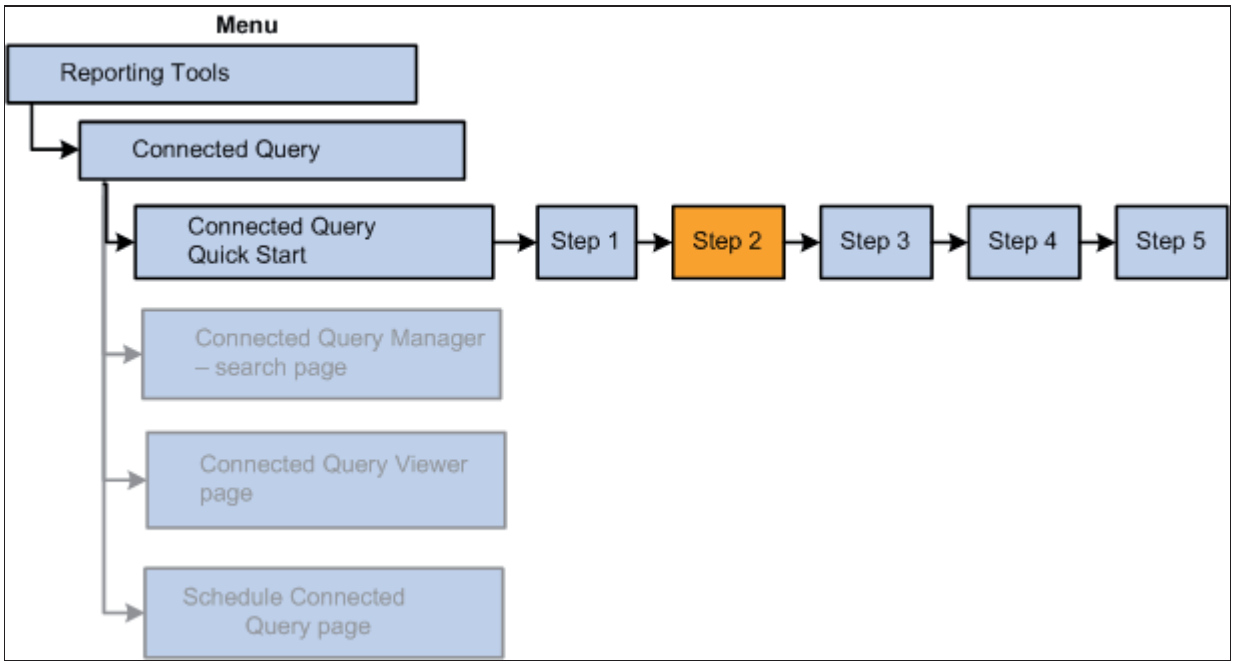
Use the Help icon to retrieve help for a specific topic.

Note. The Help icon is only available in Connected Query Quick Start.

Using Connected Query Quick Start (continued)

Connected Query Quick Start - Step 2

This diagram shows the navigation path to access the Connected Query Quick Start - Step 2 page:



Slide 207

Student Notes

Page Used to Select a Parent Query

Use this page to select an existing query to use as the parent query:

Page Name	*Navigation*
Step 2: Select a Parent Query	From the Enter a Name and Descriptive Information - Step 1 page, enter a connected query name, and click the Next button.

Elements of the Select a Parent Query Page

Parent Query Click the lookup icon to select a parent query from a list of queries for which you have been granted access.

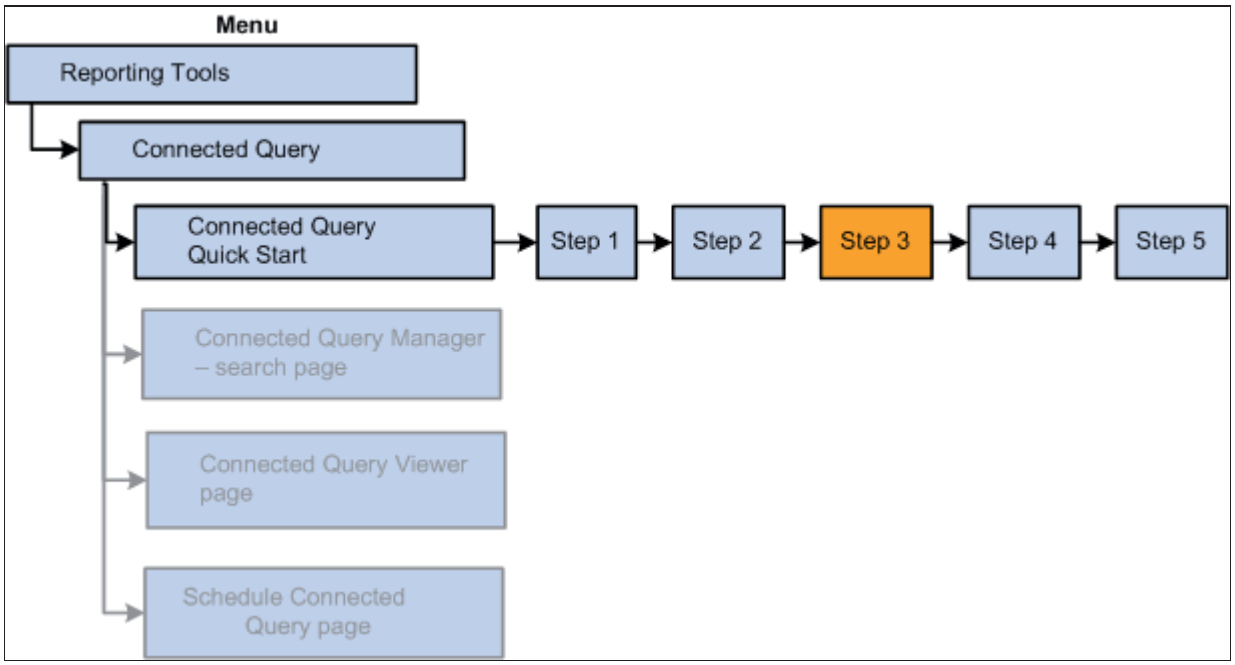
Note. You must use the query lookup icon to select a parent query and be able to access the next page.

View Query Click to open Query Manager component in read-only mode to view the query.

Using Connected Query Quick Start (continued)

Connected Query Quick Start - Step 3

This diagram shows the navigation path to access the Connected Query Quick Start - Step 3 page:



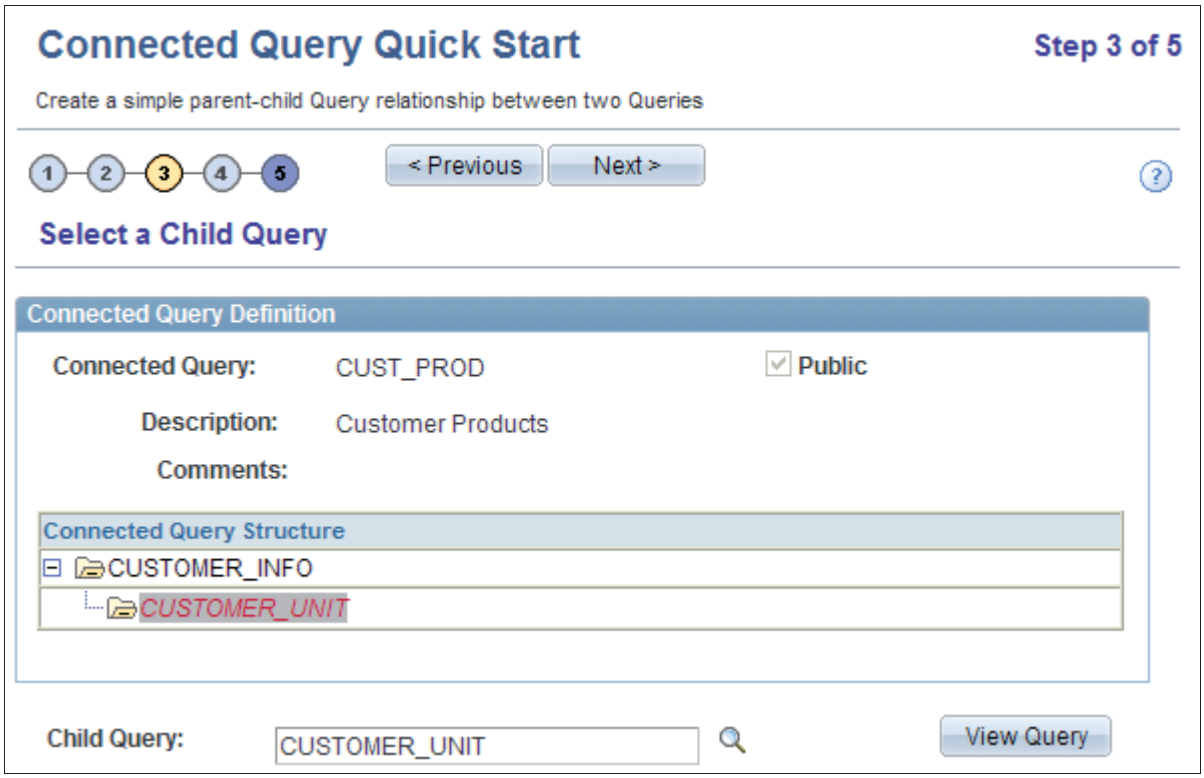
Slide 208

Student Notes

Page Used to Select a Child Query

Use this page to select an existing query to use as the child query:

Page Name	*Navigation*
Step 3: Select a Child Query	From the Select a Parent Query page, select an existing query to use as the parent query, and click the Next button.



Elements of the Select a Child Query Page

Child Query Click the lookup icon to select a child query from a list of queries for which you have been granted access.

After you select a child query, its name appears in the Connected Query Structure section.

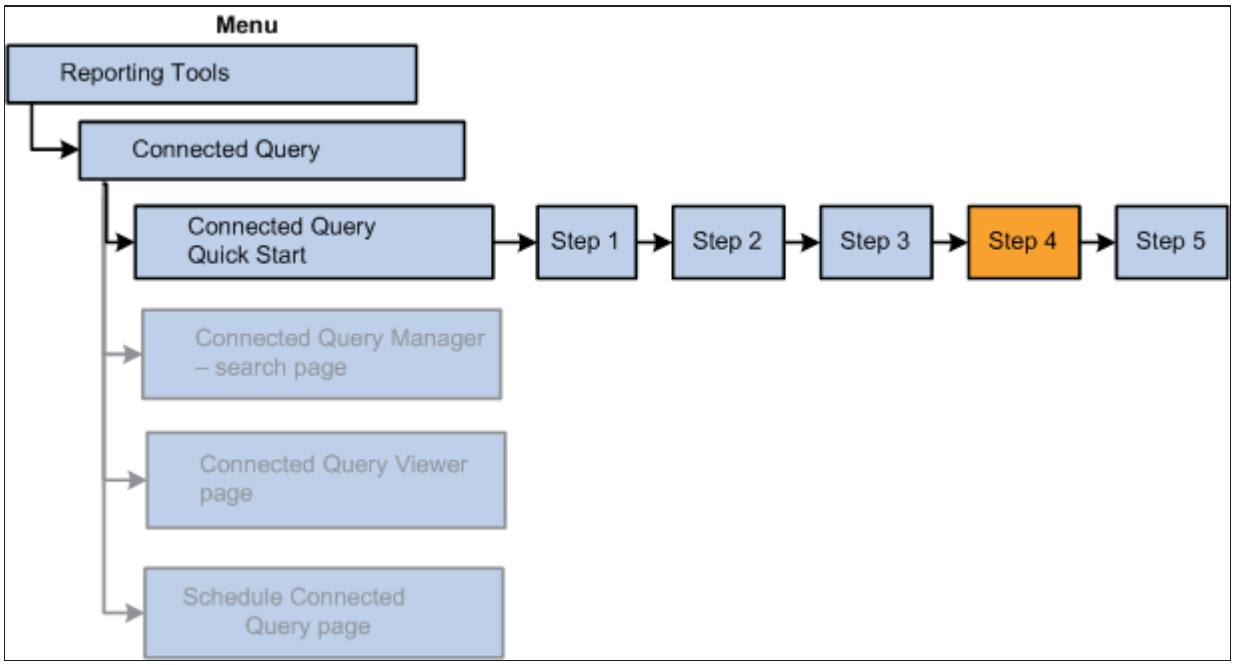
View Query This button becomes active after you select a child query.

Click to open Query Manager component in the read-only mode to view the query.

Using Connected Query Quick Start (continued)

Connected Query Quick Start - Step 4

This diagram shows the navigation path to access the Connected Query Quick Start - Step 4 page:



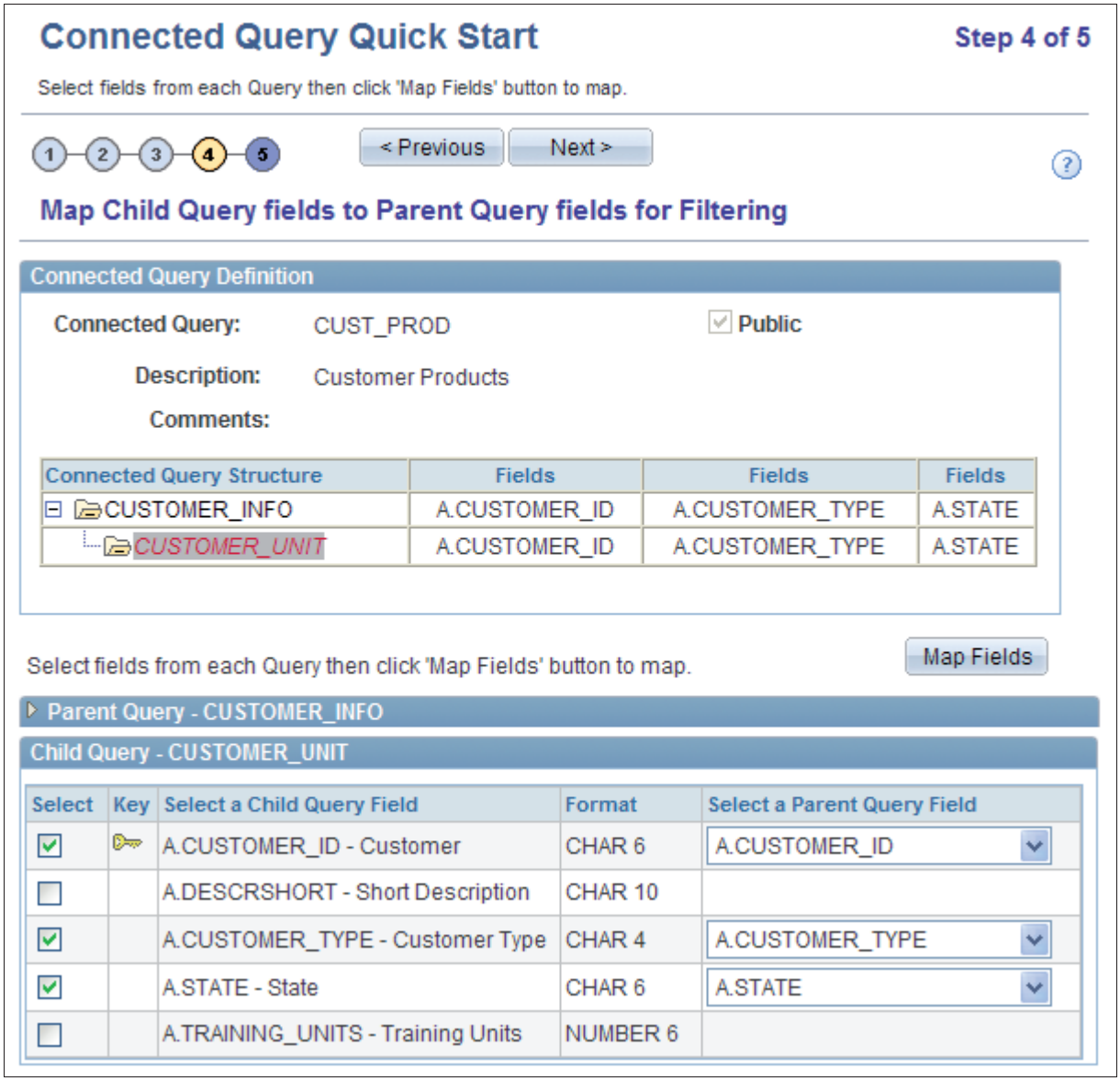
Slide 209

Student Notes

Page Used to Map Child Query Fields to Parent Query for Filtering

Use this page to map query fields from a child query to corresponding fields from a parent query:

Page Name	*Navigation*
Step 4: Map Child Query Fields to Parent Query for Filtering	From the Select a Child Query page, select an existing query to use as the child query, and click the Next button.



Map Child Query Fields to Parent Query for Filtering Page

This page displays two sets of fields coming from parent and child queries:

- The parent query section displays parent query fields and is for informational purposes only.

  This section is collapsed by default and you can optionally expand it.

- The child query section displays child query fields and enables the report developer to select one or more fields for mapping.

  For each child query field selection, a drop-down list box with possible parent query field lists appears. Those parent query field lists are filtered by compatible field data types. In some cases, Connected Query Manager uses smart thinking to map fields based on partial matching names but ultimate field matching is users' selection.

Note. While processing a member query hierarchy, Connected Query uses data in the parent query to filter data in the child query using the equal (=) operator, except the data of the EFFDT field. Therefore, if you use the EFFDT field as a related field, Connected Query uses the EFFDT operator and may unexpectedly filter the child data. For example, if the DEPARTMENT query (parent) and the EMPLOYEE query (child) both have the EFFDT field, and this EFFDT field is used for mapping, you may receive a list of employees that were assigned to a specific department and have been hired before the department became active.

If parent and child queries are not linked with some fields, each row of data for a parent query will have all rows from a child query as no filtering from a parent to child will happen. For example, *PROJECT* is a parent query and *PROJECT DESCRIPTION* is a child query. The *PROJECT* and *PROJECT DESCRIPTION* queries are linked by the PROJECT_ID field that will assume one-to-one relationship (without EFFDT field exists in a second query). Therefore, if the *PROJECT* query returns 100 rows, the *PROJECT DESCRIPTION* query will return a single row for the *PROJECT* query, and resulted data set will have 100 rows of data.

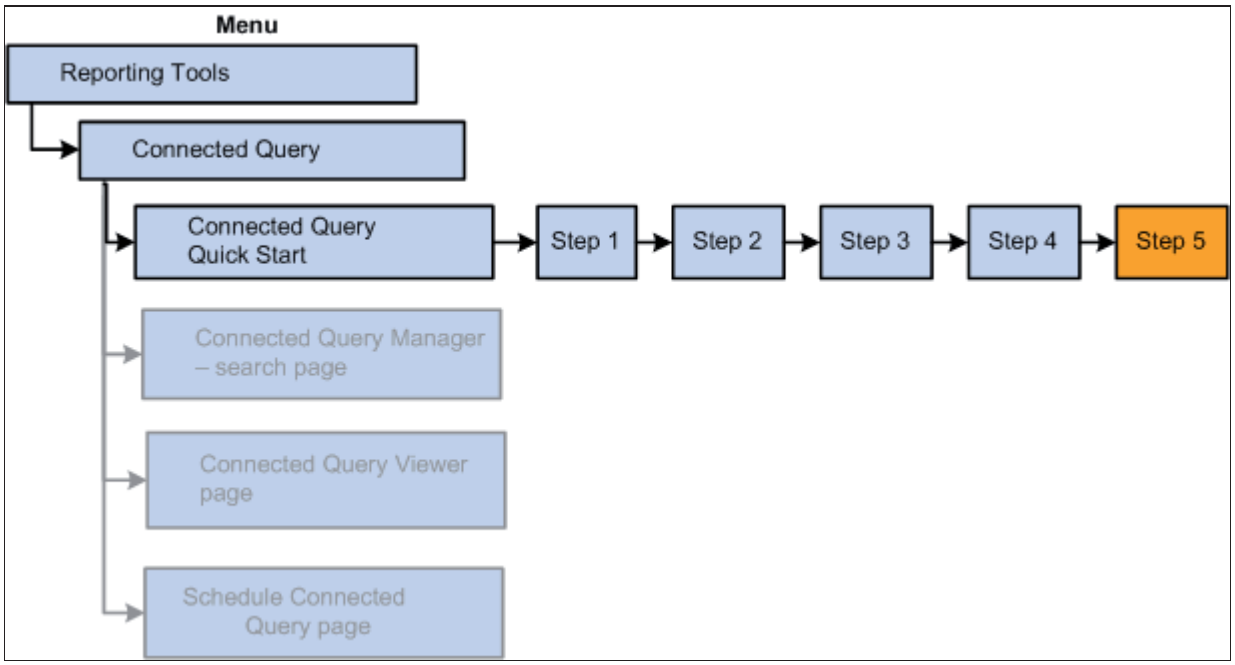
If no field is selected as a mapped field (PROJECT_ID field), Connected Query returns `100 x 100 = 10000` rows. Sometimes you do not need mapped fields; if so, the system displays a warning message when you save the connected query.

In this page, you need to click the Map Field button to update the Connected Query structure, and to view the mapping values in the Fields columns in the Connected Query Structure section.

Using Connected Query Quick Start (continued)

Connected Query Quick Start - Step 5

This diagram shows the navigation path to access the Connected Query Quick Start - Step 5 page:



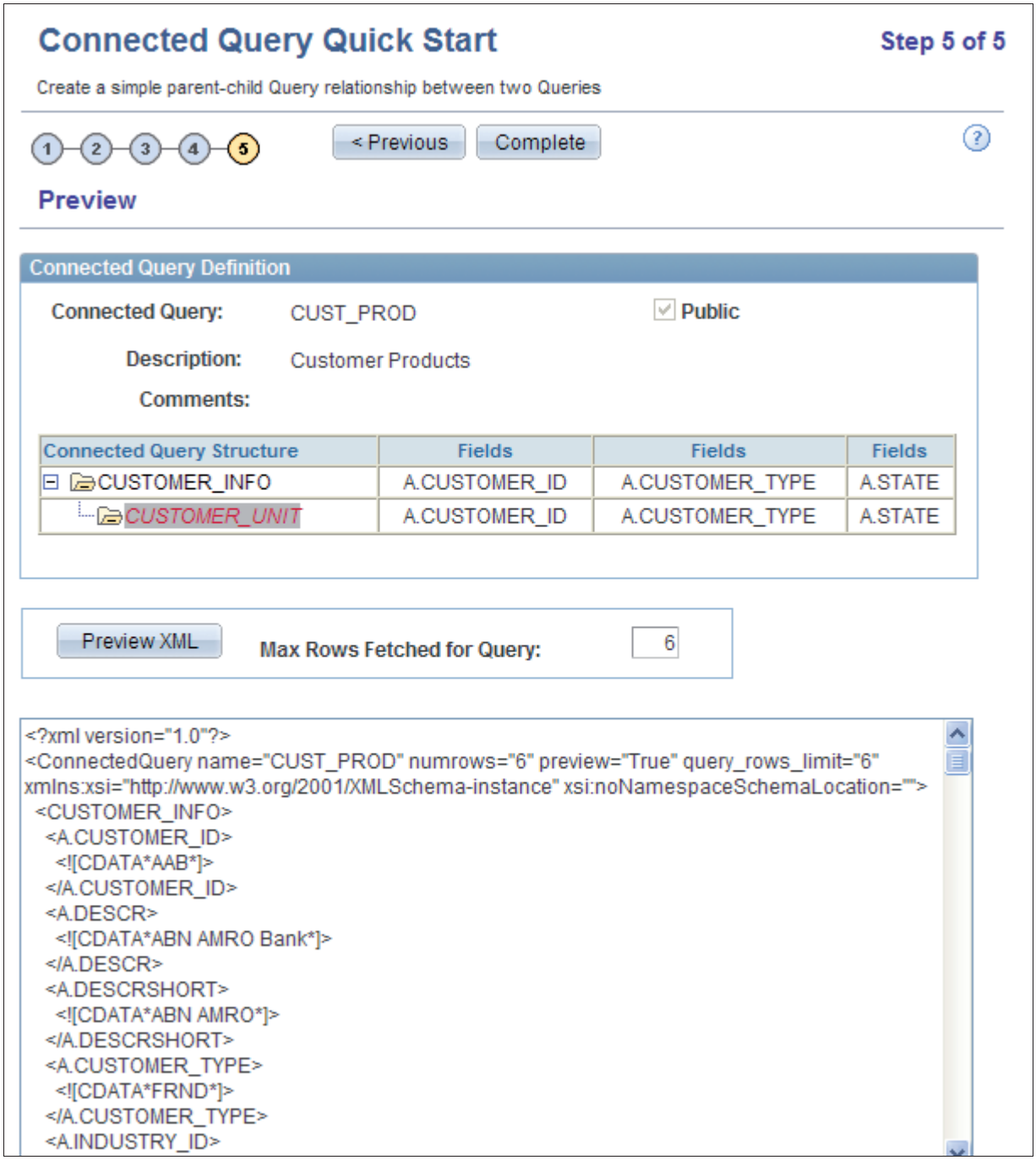
Slide 210

Student Notes

Page Used to Preview a Connected Query

Use this page to review a new connected query and preview the results by running it in a preview mode with a limited number of rows returned from each query:

Page Name	*Navigation*
Step 5: Preview	From the Map Child Query Fields to Parent Query for Filtering page, map query fields from a child query to corresponding fields from a parent query, and click the Next button.



Elements of the Preview a Connected Query Page

Preview XML

Click to display the formatted XML output for this connected query.

XML output appears in a Preview mode, in the lower section of the Preview page.

Max Rows Fetched For Query

Enter the maximum number of rows to display in the Preview mode.

Note. This field is also available in Connected Query Manager and Connected Query Viewer.
You should limit the number of rows that are being returned. The default value is *6.* If you enter *0* or clear this text box, no row limits will be imposed for a resulting XML, which can cause significant delay in getting results; and because the application runs in synchronized mode using the application server, you may get browser time-out or tuxedo time-out.

Complete

Click to transfer to the Connected Query Manager page, where you can perform enhancements and save the connected query.

Note. While working with a Connected Query Quick Start, you can modify your previous selections at any step by clicking the step number (in yellow or gray) icons. Changes in your selection could change a connection query structure.

Activity 30: Using Connected Query Quick Start

In this activity, you will review the activity overview and:

1. Enter connected query information.
2. Select a parent query and a child query.
3. Map the child query fields to the parent query for filtering.
4. Preview the results and save the connected query.
5. View the results.

Slide 211

Activity Overview

Use Connected Query Quick Start wizard to create a simple connected query (ENROLLMENT_BY_BU) with a single parent query (CRS_SESSION_BY_BU) and a single child query (STU_ENROLLMENT) linked using a set of related fields (A.COURSE and A.SESSION_NBR).

Preview and view the results using *NAM01* for the Unit prompt and *11/01/2009* for the Start Date prompt.

Note. Use the *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Entering Connected Query Information

To enter the connected query information:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Select Reporting Tools, Connected Query, Connected Query Quick Start.
3. Enter the following information:

Page Element	***Value or Status***
Connected Query	*ENROLLMENT_BY_BU*
Description	*Student Enrollment by BUS Unit*
Public	Selected

Selecting a Parent Query and a Child Query

To select a parent query and a child query:

1. From the Enter Connected Query Name and Descriptive Information page, click the Next button.
2. Use the Parent Query lookup icon to search and select the *CRS_SESSION_BY_BU* parent query.
3. Click the Next button.
4. Use the Child Query lookup icon to search and select the *STU_ENROLLMENT* child query.

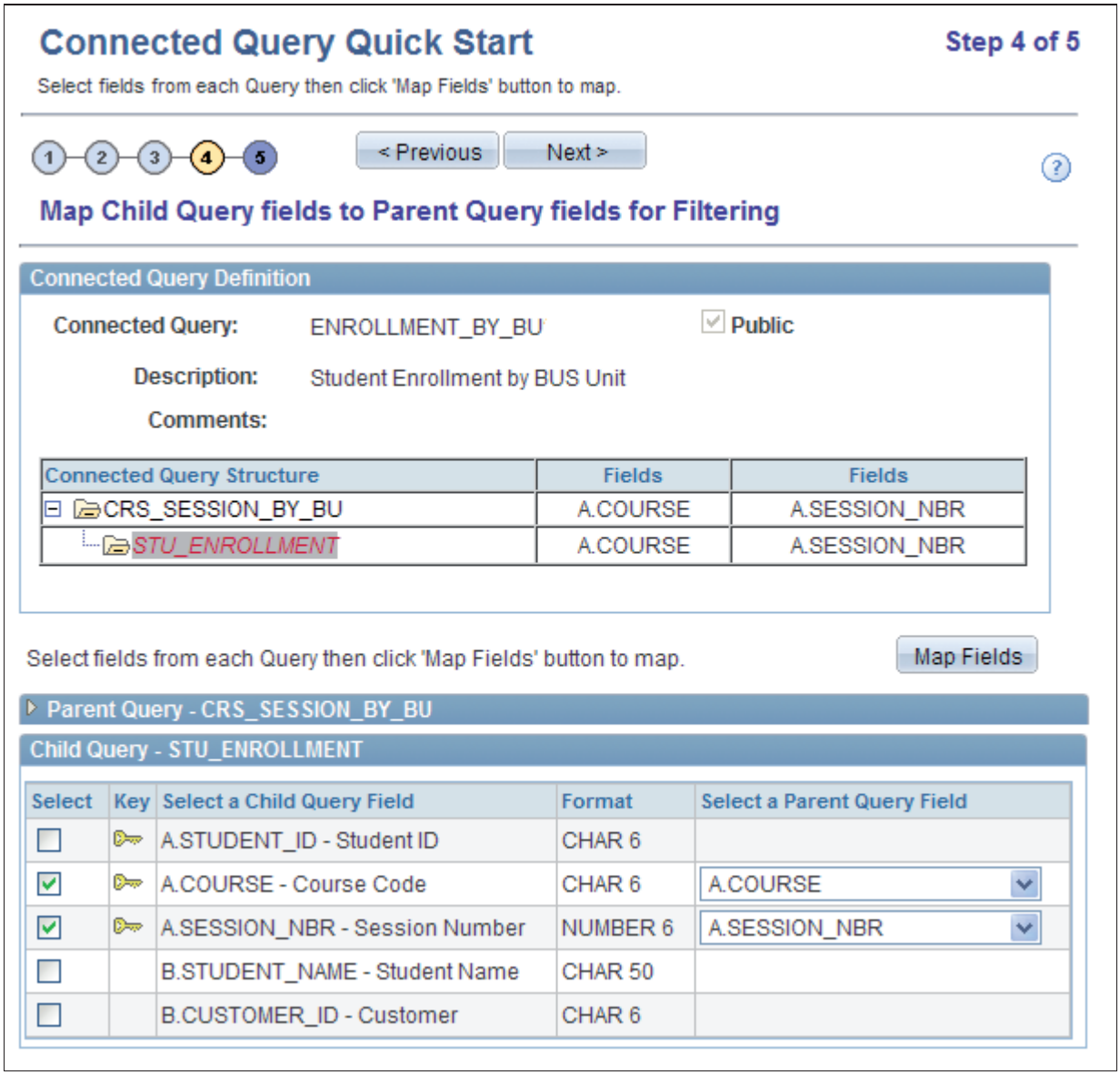
Mapping Child Query Fields to Parent Query for Filtering

To map the child query fields to the parent query for filtering:

1. From the Select a Child Query page, click the Next button.
2. Select the *A.COURSE* child query field, and confirm that it is mapped to the *A.COURSE* parent query field.
3. Select the *A.SESSION_NBR* child query field, and confirm that it is mapped to the *A.SESSION_NBR* parent query field.
4. Click the Map Fields button.

Results

This example shows the Map Child Query Fields to Parent Query Fields for Filtering page:



Previewing the Results and Saving the Connected Query

To preview the results and save the query:

1. From the Map Child Query Fields to Parent Query Fields for Filtering page, click the Next button.
2. Click the Preview XML button, and enter the following information for the prompts:

Page Element	***Value or Status***
Unit	*NAM01*
Start Date	*11/01/2009*

3. Click the OK button to return to the Preview page.

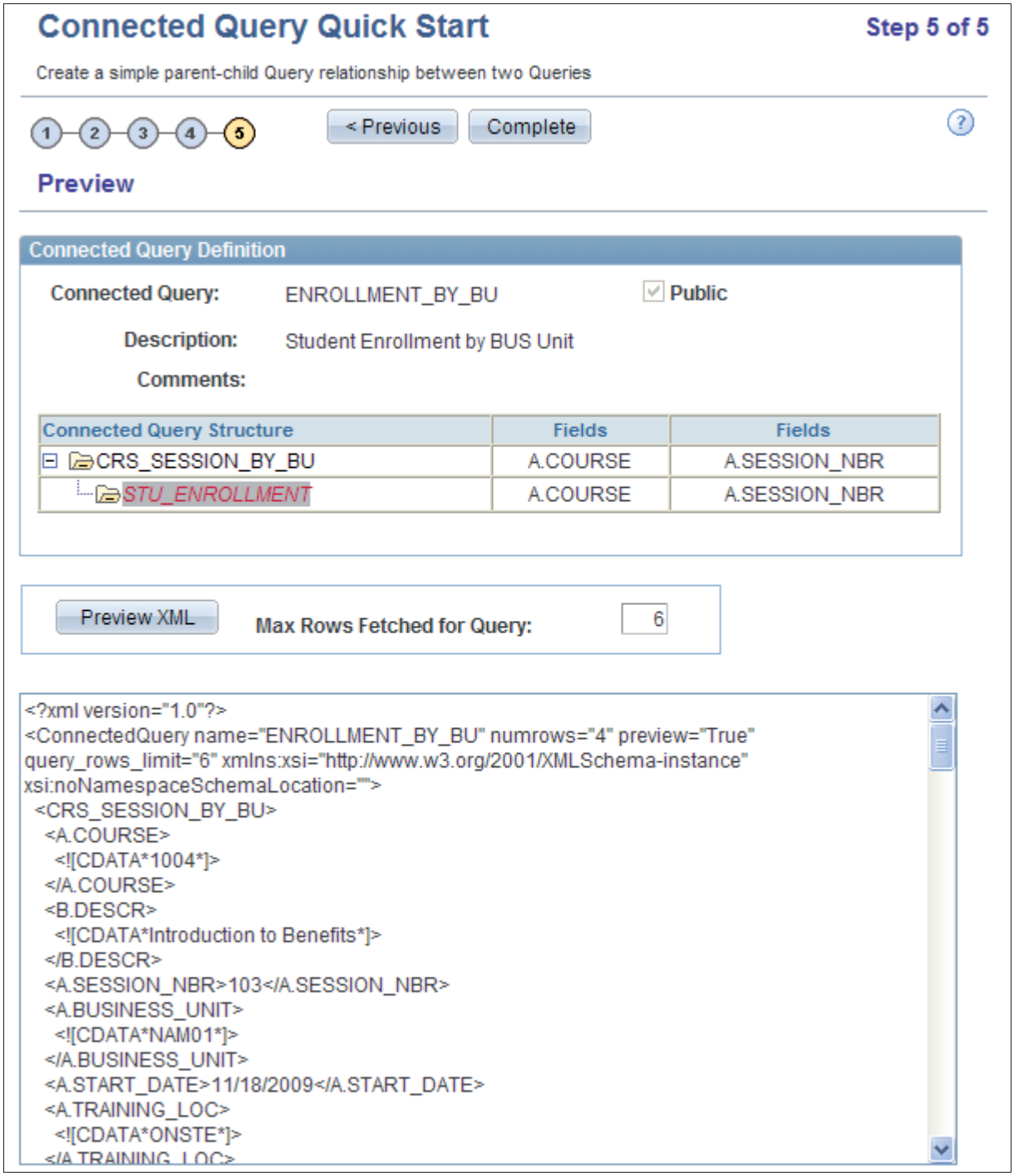
4. View the XML in the Preview XML section, and click the Complete button.

   The Connected Query Manager page appears.

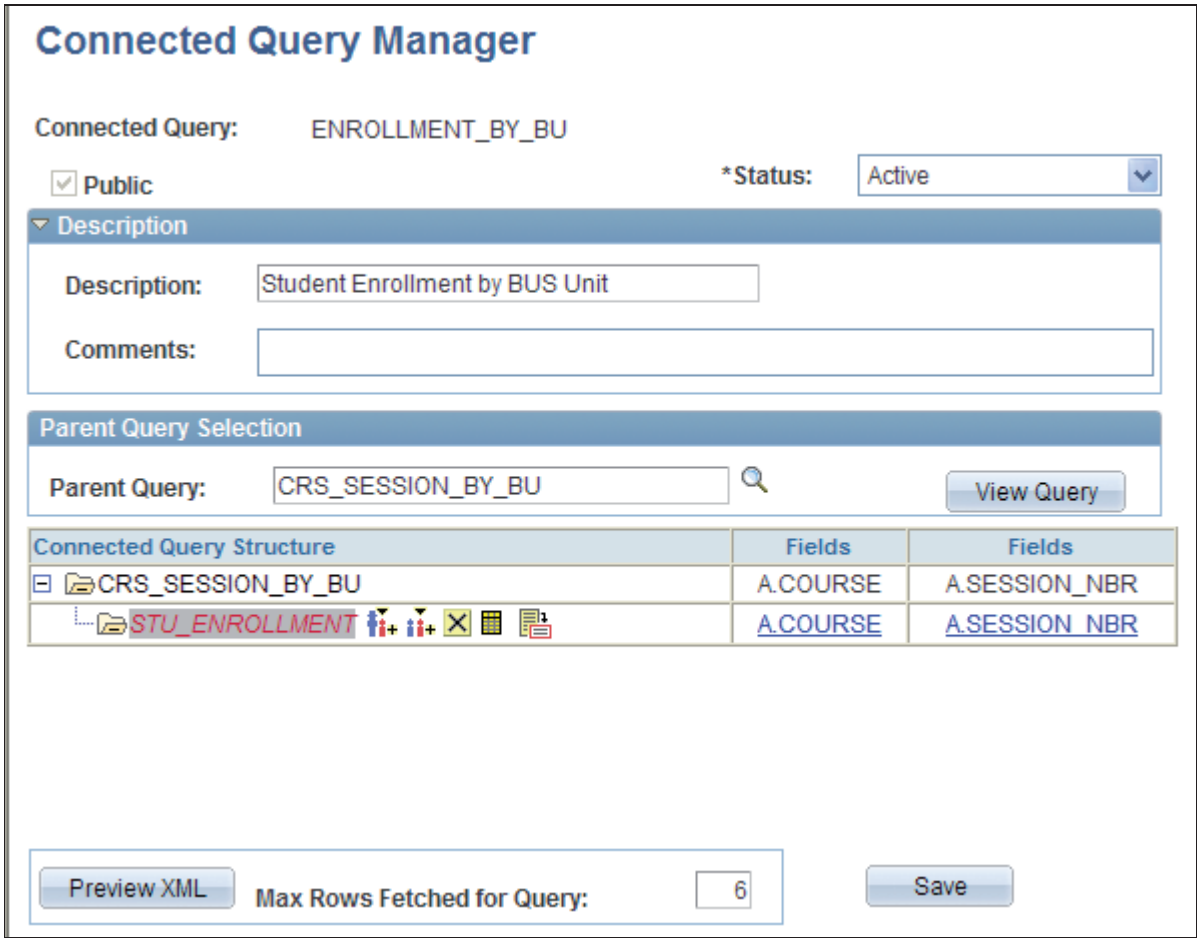
5. Save the connected query.

Results

This example show the Preview page with structured XML:



This example shows the Connected Query Manager displaying the ENROLLMENT_BY_BU connected query:



Viewing the Results

To view the results:

1. In the Connected Query Manager page, click the Preview XML button.
2. Enter the following information:

Page Element	***Value or Status***
Unit	*NAM01*
Start Date	*11/01/2009*

3. Click the OK button.

   Connected Query displays the XML report in a separate window.

4. Compare your outputs with the following results.

Results

This example shows the ENROLLMENT_BY_BU connected query in structured XML format:

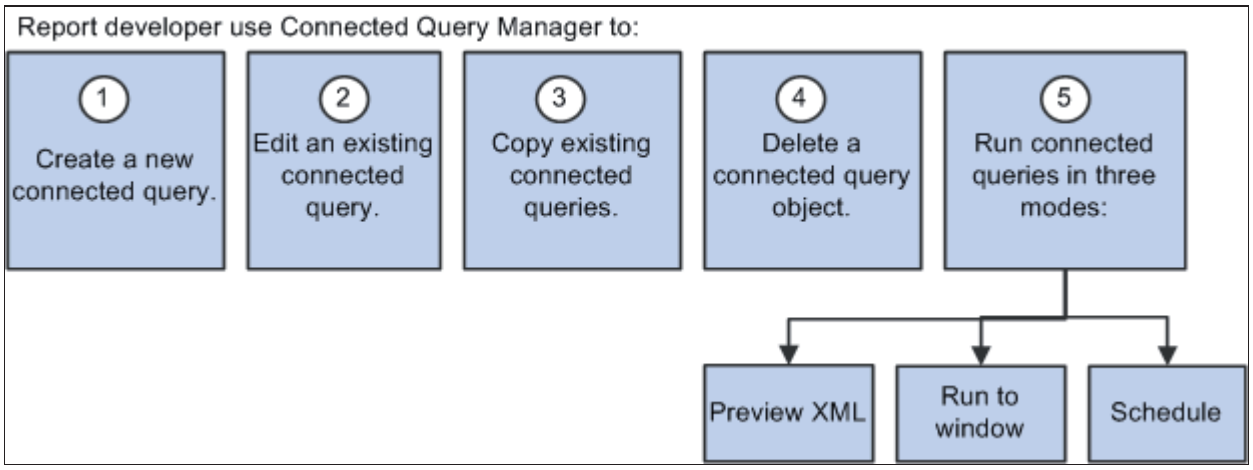
```
<?xml version="1.0" ?>
- <ConnectedQuery name="ENROLLMENT_BY_BU" numrows="4" preview="True" query_rows_limit="6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="">
  - <CRS_SESSION_BY_BU>
    - <A.COURSE>
        <![CDATA[ 1004  ]]>
      </A.COURSE>
    - <B.DESCR>
        <![CDATA[ Introduction to Benefits  ]]>
      </B.DESCR>
      <A.SESSION_NBR>103</A.SESSION_NBR>
    - <A.BUSINESS_UNIT>
        <![CDATA[ NAM01  ]]>
      </A.BUSINESS_UNIT>
      <A.START_DATE>11/18/2009</A.START_DATE>
    - <A.TRAINING_LOC>
        <![CDATA[ ONSTE  ]]>
      </A.TRAINING_LOC>
    </CRS_SESSION_BY_BU>
  - <CRS_SESSION_BY_BU>
    - <A.COURSE>
        <![CDATA[ 1023  ]]>
      </A.COURSE>
    - <B.DESCR>
        <![CDATA[ Query/Crystal Power Reporting  ]]>
      </B.DESCR>
      <A.SESSION_NBR>231</A.SESSION_NBR>
    - <A.BUSINESS_UNIT>
        <![CDATA[ NAM01  ]]>
      </A.BUSINESS_UNIT>
      <A.START_DATE>12/10/2009</A.START_DATE>
    - <A.TRAINING_LOC>
        <![CDATA[ TEA  ]]>
      </A.TRAINING_LOC>
    </CRS_SESSION_BY_BU>
```

This concludes the activity. Please do not continue.

Using Connected Query Manager

Connected Query Manager

This diagram shows the usages of the Connected Query Manager page:



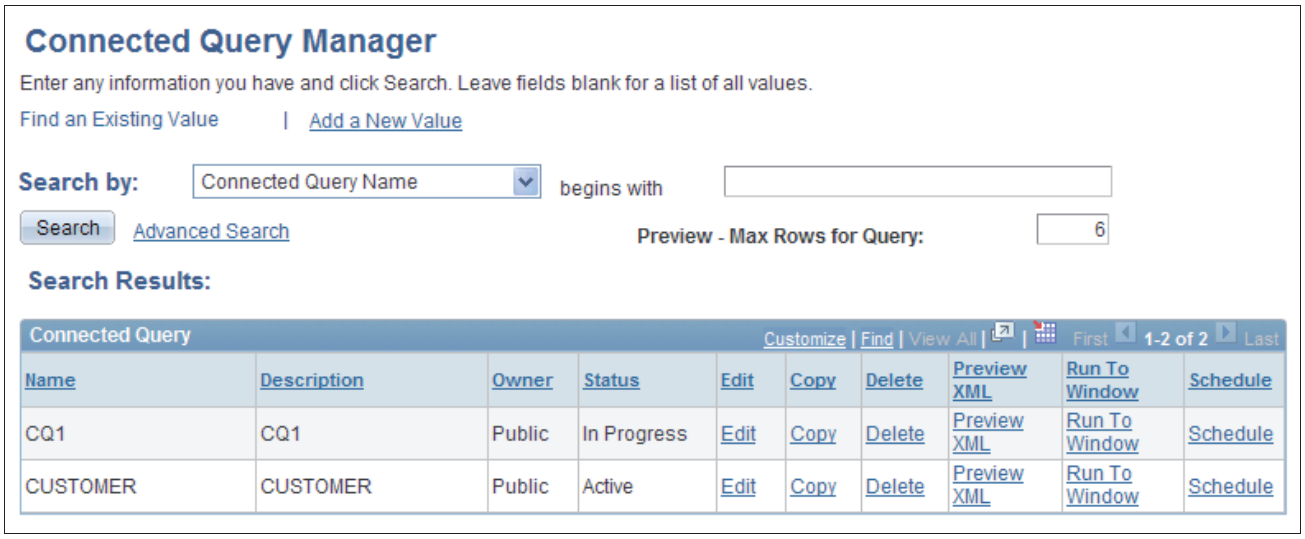
Slide 212

Student Notes

Page Used to Create and Maintain a Connected Query

Use this page to edit a connected query, copy an existing connected query, delete a connected query, preview results of a connected query, schedule a connected query to run immediately with results displayed in a separate window, and schedule a connected query to run at the preset time:

Page Name	*Navigation*
Connected Query Manager search	Reporting Tools, Connected Query, Connected Query Manager



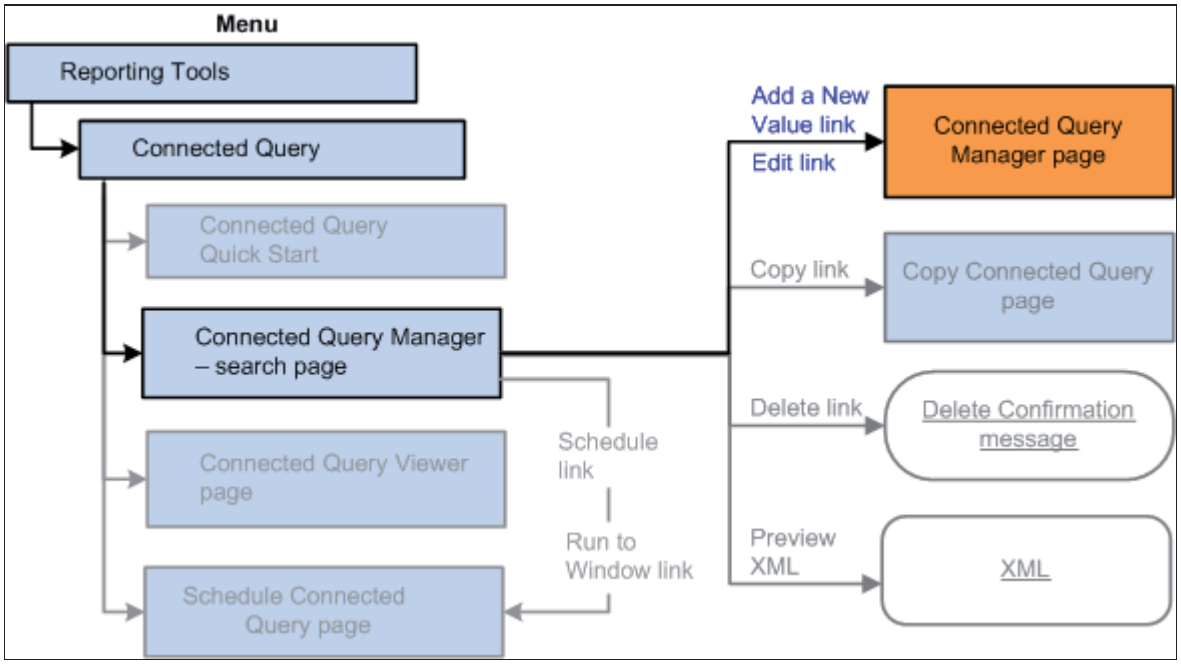
Connected Query Manager

Connected Query structure consists of a set of PeopleSoft queries that are linked by related fields. There are no limits for the number of fields being linked between two queries. Parent and child pairs of queries can be linked together using sets of correlated fields, though those linkages are not required.

Note. XML is the only output format currently supported.

Using Connected Query Manager (continued)

This diagram shows the navigation path to access the Connected Query Manager page:



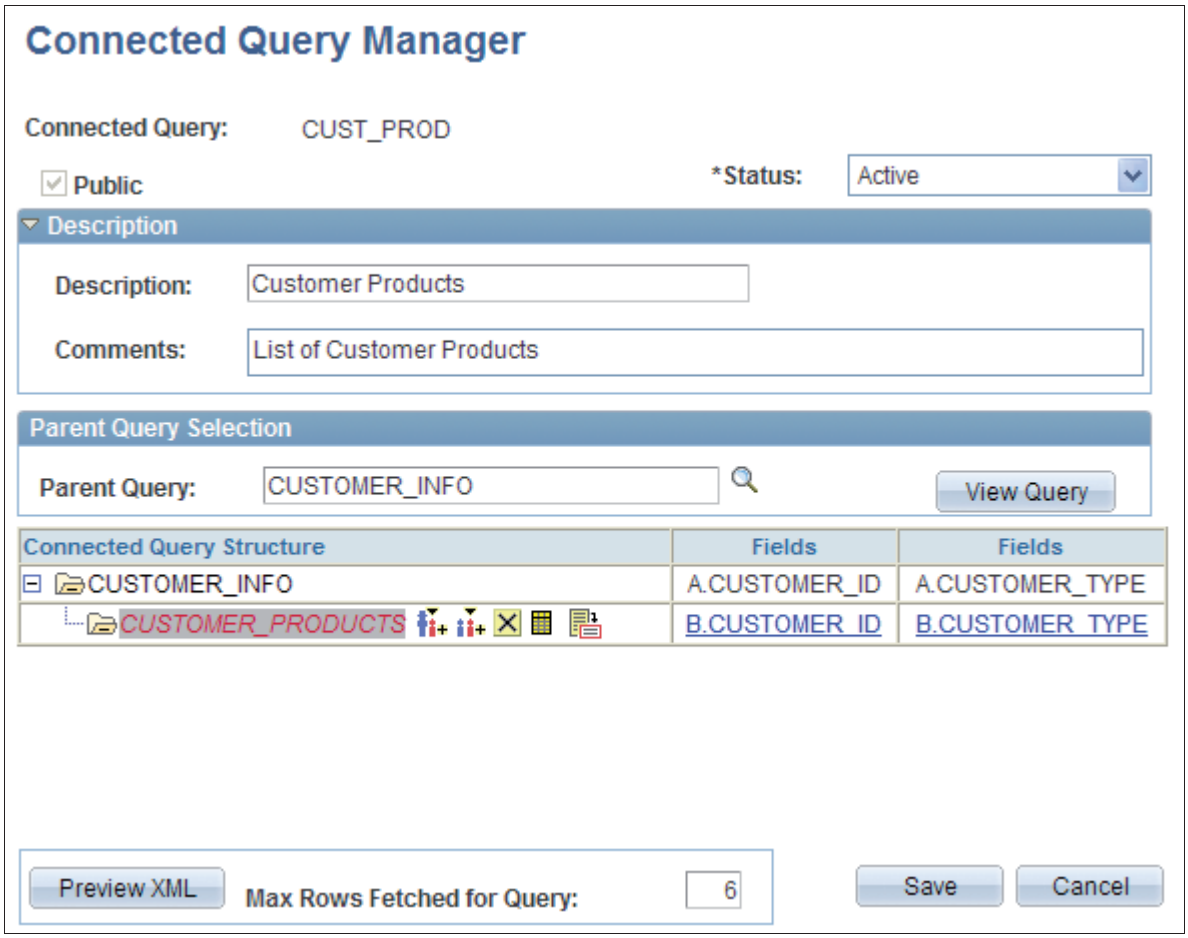
Slide 213

Student Notes

Page Used to Create and Maintain Connected Queries

Use this page to create and maintain connected queries:

Page Name	*Navigation*
Connected Query Manager	1. Select Reporting Tools, Query, Query Manager. 2. Search for and click the Edit link to open an existing connected query. Alternatively, click the Add a New Value link to create a new connected query.



Elements of the Connected Query Manager Page

The elements of the Connected Query Manager page are:

Status Select the status of the connected query.

Note. The *Inactive* status indicates that a connected query does not pass a validation routine. When a connected query is not valid, this status is set automatically during the Save process.

Select the *Active* option to enable the user to schedule a connected query.

Select the *In Progress* option to indicate that this query can be previewed in Connected Query Manager, but can not be scheduled. The *In Progress* status is useful when a connected query is not ready for end user to view using the Connected Query Viewer page.

Description Enter a description for the connected query.

If this field is blank, Connected Query populates the description using the connected query name.

Parent Query After you select a parent query, the Connected Query Structure section appears with image icons, which enables you to continue building a connected query. You can either type a parent query name in the Parent Query field, or click the lookup icon next to the Parent Query field to access a query selection page and select one query from an existing query list.

View Query Click to open the selected query in read-only mode in the Query Manager component.

You can preview a connected query at any point without having previously saved it.

Preview XML Click to preview the XML report in a separate window.

Steps Used to Create a New Connected Query Using Connected Query Manager

To create a connected query using Connected Query Manager:

1. Select Reporting Tools, Connected Query, Connected Query Manager.
2. Select the Add a New Value link.
3. Enter a name in the Connected Query field, and define query ownership using the Public check box.
4. Select the status of the connected query using the Status drop-down list box.
5. Optionally, enter description and comments for the connected query using the Description and Comments fields.
6. Select a parent query using the Parent Query field.
7. Optionally, click the View Query button to view the selected query.
8. Optionally, continue building your connected query by adding child queries, linking fields between parent and child queries, and so on.
9. Save your connected query.

   Because connected query can be created with one parent query presented, you can save your connected query and use it as a new connected query.

Steps Used to Edit an Existing Connected Query using Connected Query Manager

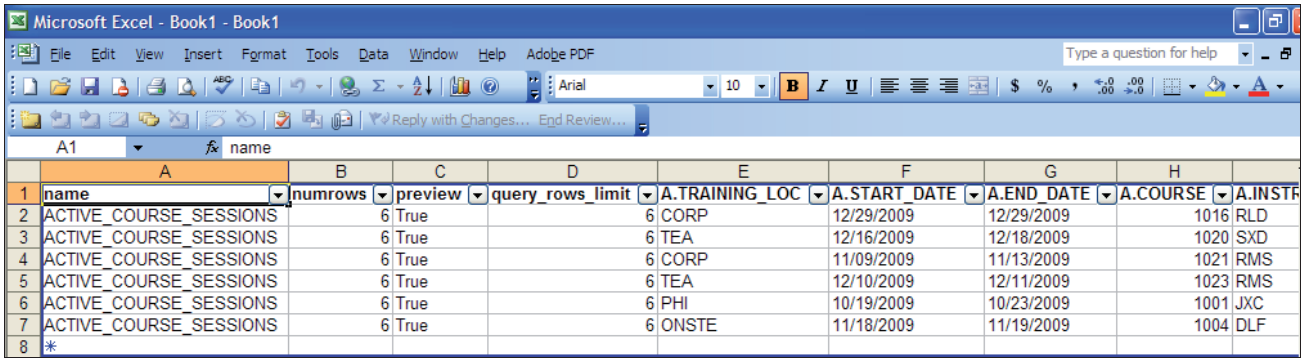
To edit an existing connected query using Connected Query Manager:

1. Access the Connected Query Manager page by selecting Reporting Tools, Connected Query, Connected Query Manager.
2. Enter your search criteria and click the Search button.
3. In the Connected Query Manager search page, click the Edit link on the row of the connected query that you want to edit.
4. Optionally, change the connected query structure, edit description fields, and change the connected query status.

Viewing Output XML Reports in Microsoft Excel

You can open connected query's output XML reports in Microsoft Excel and use all Excel functionality with XML data.

This example shows an output XML report in Microsoft Excel:

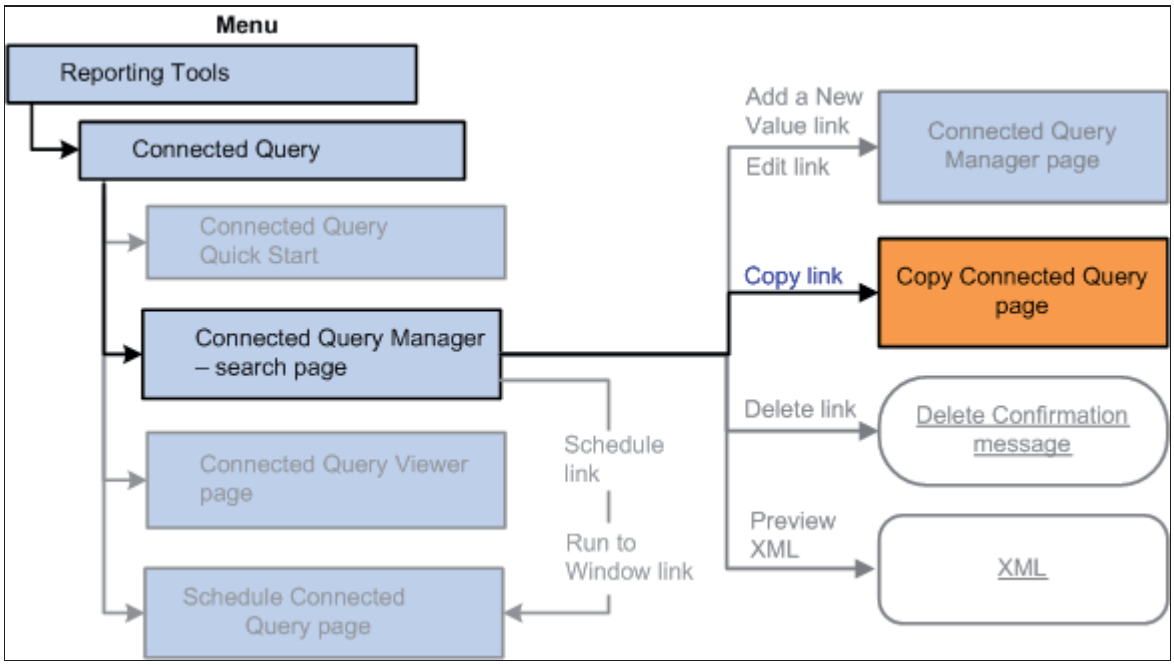


	A	B	C	D	E	F	G	H	
1	name	numrows	preview	query_rows_limit	A.TRAINING_LOC	A.START_DATE	A.END_DATE	A.COURSE	A.INSTR
2	ACTIVE_COURSE_SESSIONS	6	True	6	CORP	12/29/2009	12/29/2009	1016	RLD
3	ACTIVE_COURSE_SESSIONS	6	True	6	TEA	12/16/2009	12/18/2009	1020	SXD
4	ACTIVE_COURSE_SESSIONS	6	True	6	CORP	11/09/2009	11/13/2009	1021	RMS
5	ACTIVE_COURSE_SESSIONS	6	True	6	TEA	12/10/2009	12/11/2009	1023	RMS
6	ACTIVE_COURSE_SESSIONS	6	True	6	PHI	10/19/2009	10/23/2009	1001	JXC
7	ACTIVE_COURSE_SESSIONS	6	True	6	ONSTE	11/18/2009	11/19/2009	1004	DLF
8	*								

Using Connected Query Manager (continued)

Copying Connected Queries

This diagram shows the navigation path to access the Copy Connected Query page:



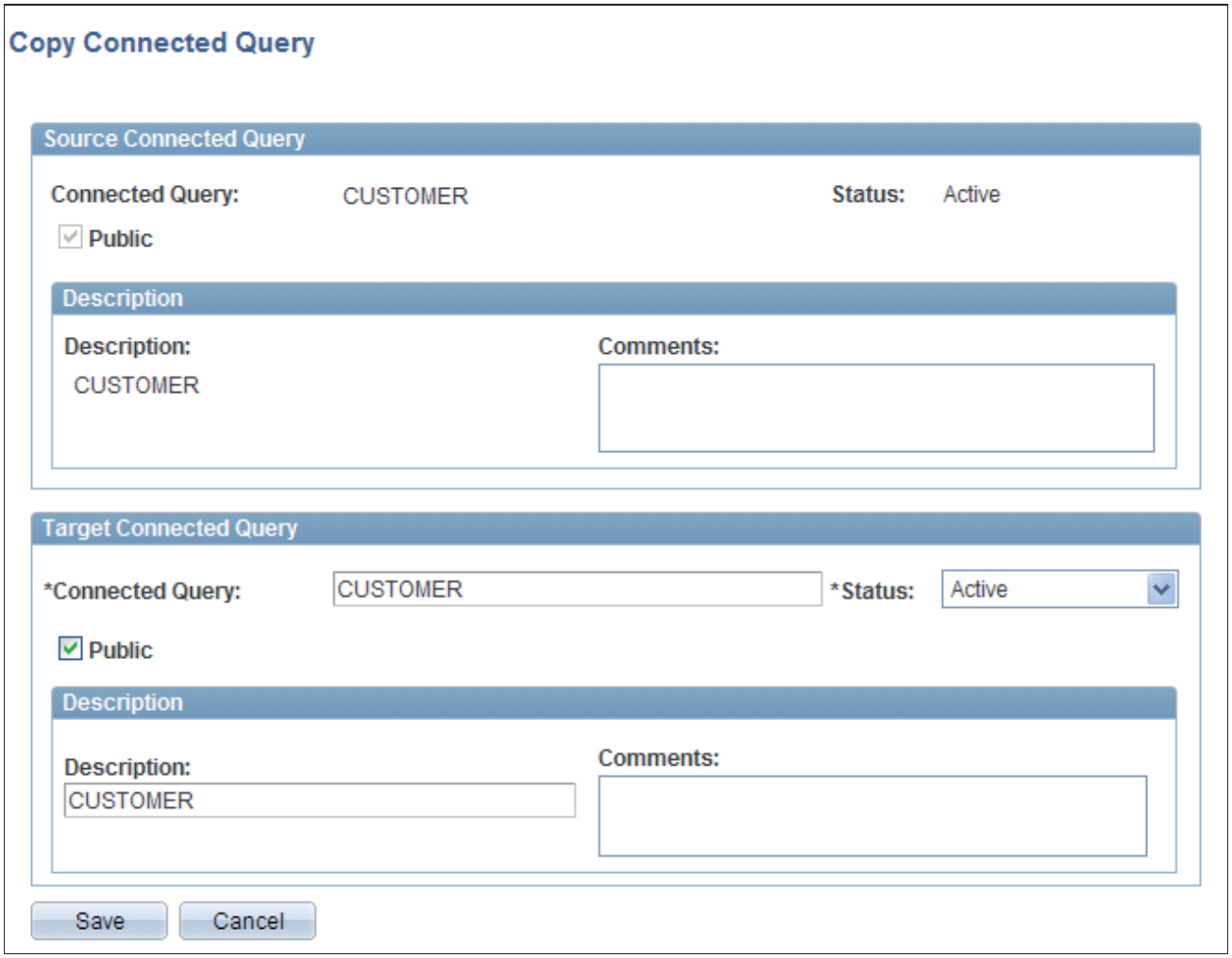
Slide 214

Student Notes

Page Used to Copy Connected Queries

Use this page to copy a connected query:

Page Name	*Navigation*
Copy Connected Query	1. Select Reporting Tools, Connected Query, Connected Query Manager. 2. Click the Copy link for the connected query to copy.



Steps Used to Copy a Connected Query

To copy a connected query:

1. Select Reporting Tools, Connected Query, Connected Query Manager.
2. Enter your selection criteria and click the Search button.
3. On the Connected Query Manager page, click the Copy link next to the connected query that you want to copy.

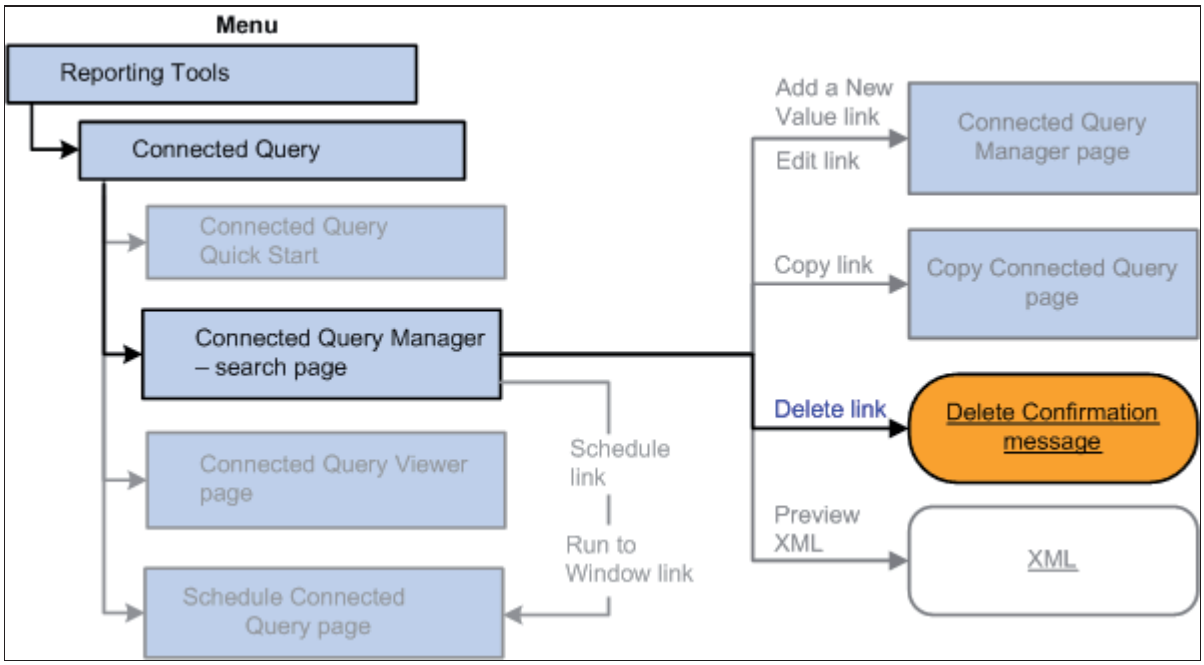
   The Copy Connected Query page appears.
4. Save a copy of the connected query by changing the connected query name in the Target section with a new name.
5. Optionally, change ownership, descriptions, or the status of the query.

6. Click the OK button.

Using Connected Query Manager (continued)

Deleting Connected Queries

This diagram shows the navigation path to delete a connected query:

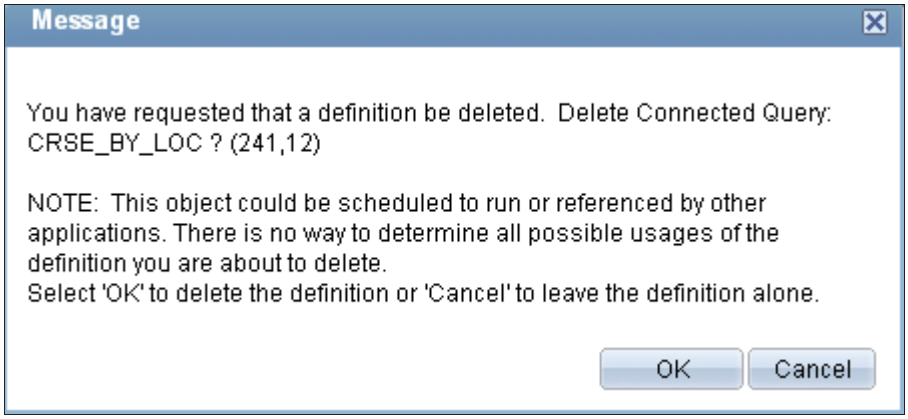


Slide 215

Student Notes

Example: Deleting Confirmation Message

This example shows the confirmation message after you click the Delete link on the Connected Query Manager search page:



Steps Used to Delete a Connected Query

To delete a connected query:

1. Select Reporting Tools, Connected Query, Connected Query Manager.
2. Click the Delete link next to the connected query that you want to delete.

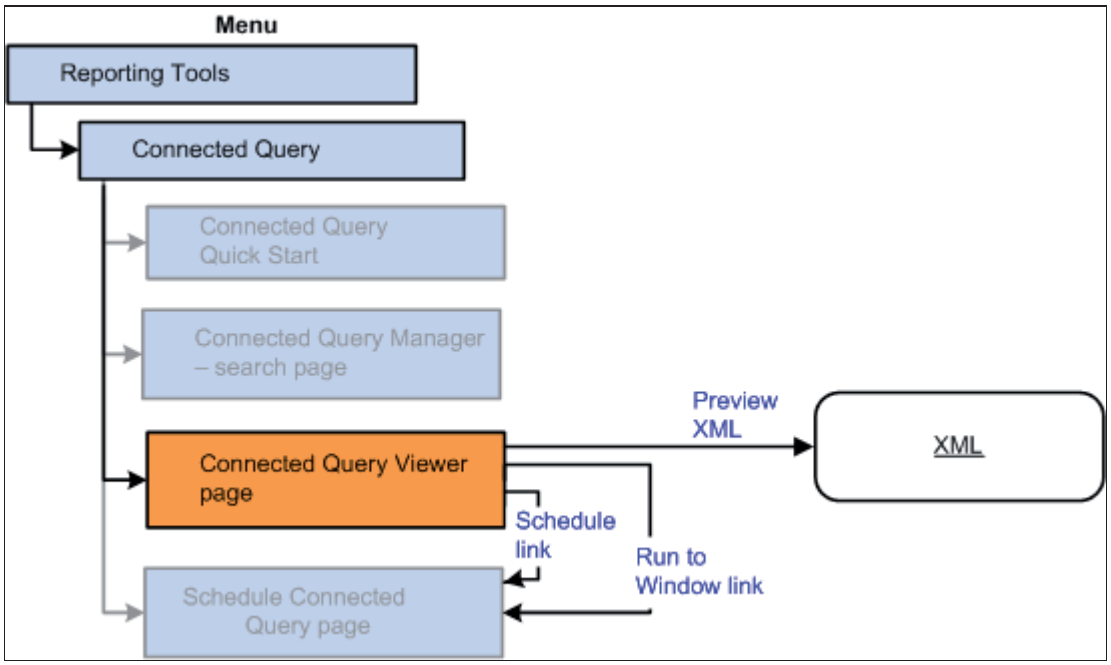
   A confirmation message appears asking if you want to delete the existing connected query.
3. Click the OK button to continue and delete the connected query.

   Alternatively, click the Cancel button to cancel the deletion and return to the Connected Query Manager page.

Using Connected Query Viewer

Accessing Connected Query Viewer

This diagram shows the navigation path to access the Connected Query Viewer page:



Slide 216

Student Notes

Connected Query Viewer

Connected Query Viewer is a read-only version of Connected Query Manager. It enables security administrators to limit users with a role of Connected Query User (permission list PTPT2800) to search for and view connected queries that have a status of *Active.*

Connected query users can view results of running connected queries using these three methods:

- Preview the XML.
- Schedule a connected query to run immediately with results displayed in a separate window.

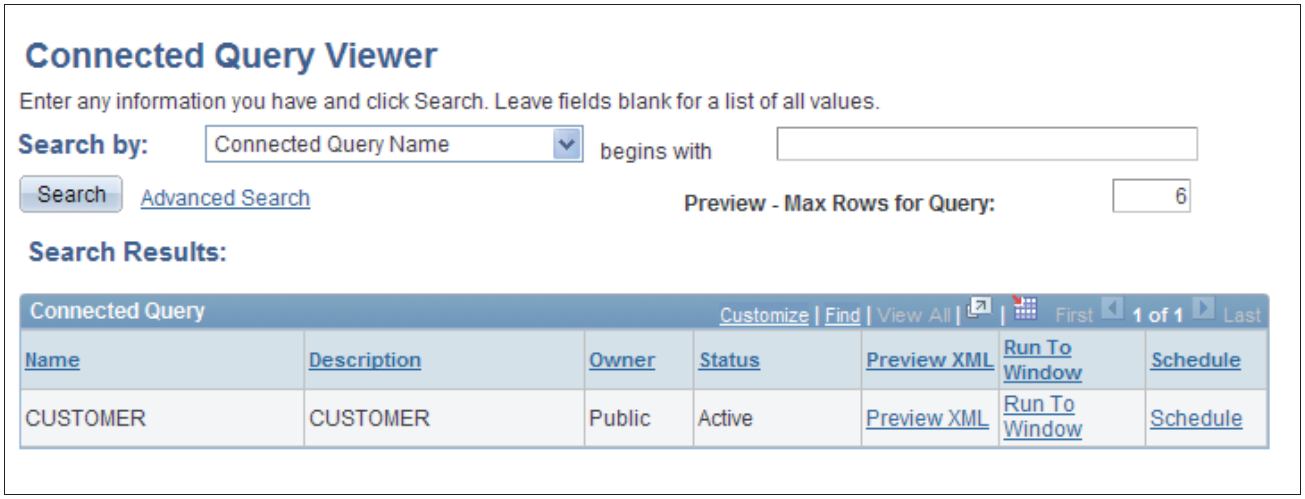
Note. To use the Run To Window functionality, the REN server must be running.
Since XML output is delivered into a separate browser window and into Report Manager, you can view XML results in Report Manager without rerunning a connected query.

- Schedule the connected query to run at a later time.

Page Used to View the Results of a Running Connected Query

Use this page to view the results of a running connected query in the Preview XML, Run to Window, or Schedule modes:

Page Name	*Navigation*
Connected Query Viewer	Select Reporting Tools, Connected Query, Connected Query Viewer.



Search Click to search for a connected query using the basic search function.

Combined with only the begins with condition, the basic search enables you to select the following *Search by* criteria: Connected Query Description, Connected Query Name, Connected Query Status, and Owner.

Preview - Max Rows For Query Enter a number of rows to be returned from each query in a preview mode.

You should limit the number of rows that are being returned. The default value is 6. If you enter 0 or clear this text box, no row limits will be imposed for a resulted XML, which can cause a significant delay in getting results; and because the application runs in a synchronized mode using application server, you may get browser time-out or tuxedo time-out errors.

Preview XML Click a Preview XML link for the selected connected query to view its results in a new browser window.

The Preview XML action uses the Application Server to generate and configure XML data, thus you have to wait for XML data to be available. To keep working while system generates and configures XML data, use the Run to Window link, which does not limit number of rows returned from each query.

If the connected query has prompts, a prompt window appears after you click the Preview XML button.

Run to Windows Click a Run To Windows link for the selected connected query to schedule that query to be run immediately using PeopleSoft Process Scheduler, and then view its XML results in a new browser window.

Schedule

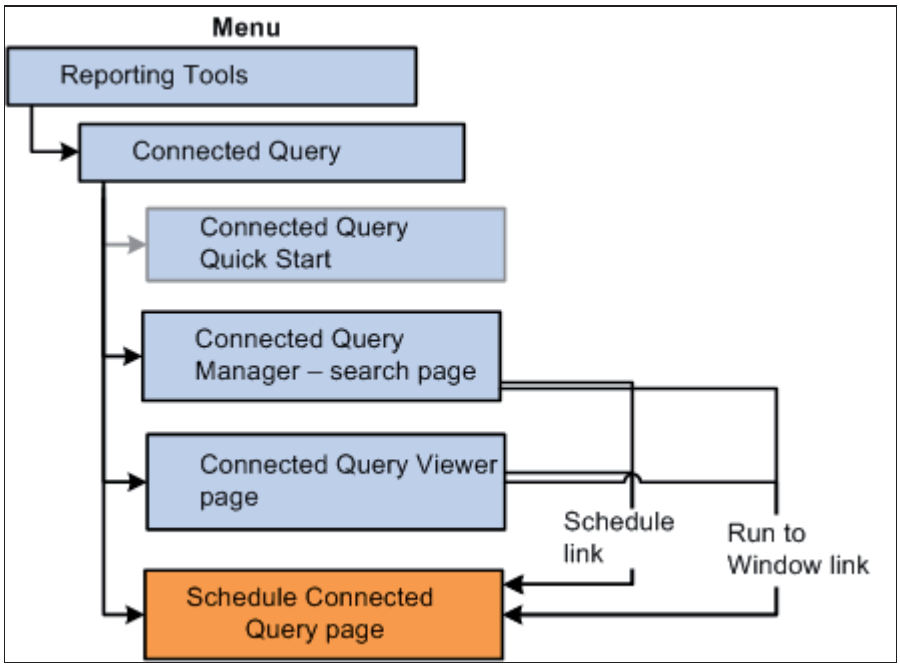
Click the Schedule link that is associated with the connected query.

Connected Query Viewer interacts with PeopleSoft Process Scheduler to enable you to schedule connected queries. You can submit requests to schedule a connected query, check the status of your request using Process Monitor, and view connected query results using Report Manager.

Using Connected Query Scheduler

Accessing Connected Query Scheduler

This diagram shows the navigation paths to access the Schedule Connected Query page:



Slide 217

Student Notes

Connected Query Scheduler

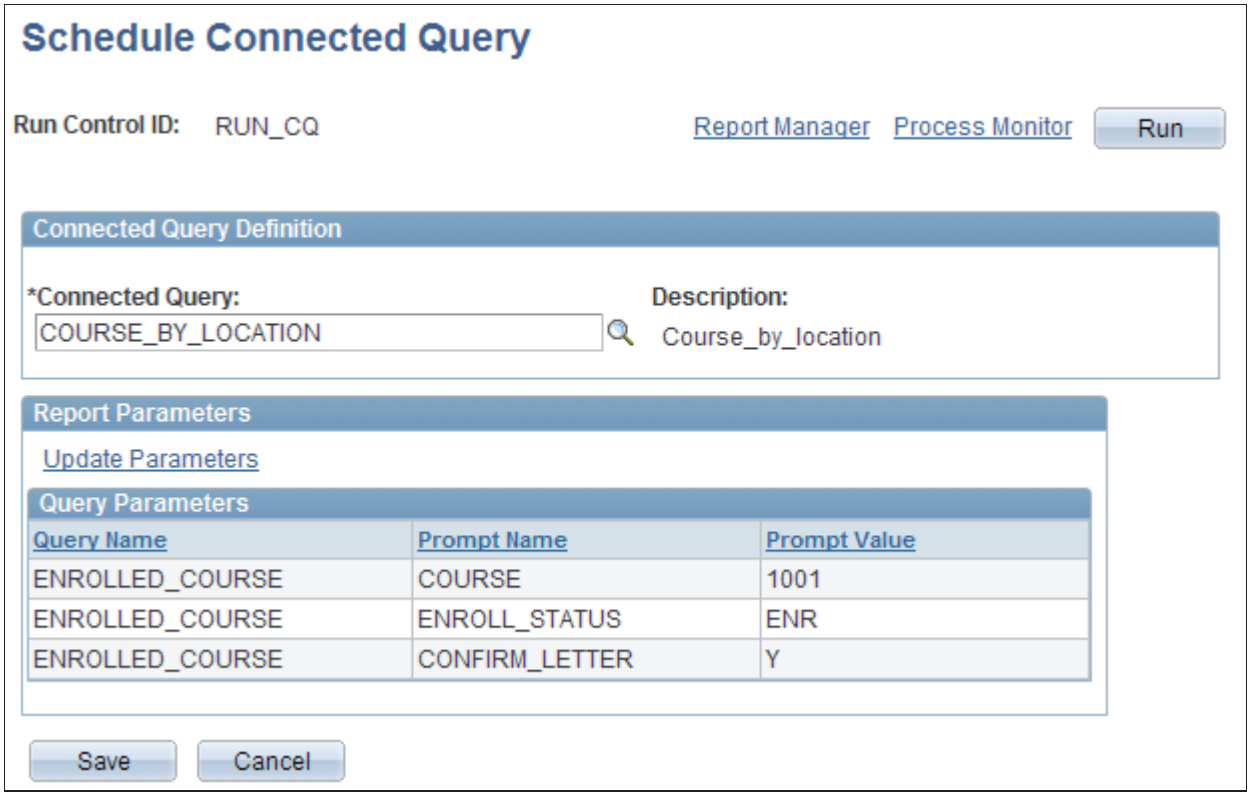
You can schedule a connected query in three ways:

- Selecting Reporting Tools, Connected Query, Connected Query Scheduler.
- Clicking the Schedule link in the Connected Query Manager search page or in the Connected Query Viewer search page.
- Clicking the Run To Window link in the Connected Query Manager search page or the Connected Query Viewer search page.

Page Used to Schedule a Connected Query

Use this page to schedule a connected query to run in the PeopleSoft Process Scheduler:

Page Name	*Navigation*
Schedule Connected Query	Reporting Tools, Connected Query, Schedule Connected Query



Elements of the Schedule Connected Query Page

Update Parameters Click to update the runtime prompt values of connected query.

Run Click to access the Process Scheduler Request page, where you can specify variables, such as where a process runs and in what format the process output is generated. The values for output type and output format appear by default per the report definition and can be changed if the report definition allows it.

Activity 31: Using Connected Query Manager and Connected Query Viewer

In this activity, you will review the activity overview and:

- Create a connected query using Connected Query Manager.
- View a connected query in Connected Query Viewer.

Slide 218

Activity Overview

Use Connected Query Manager to create the *ACTIVE_COURSE_SESSIONS* connected query. Set the SESSION_SCHEDULE query as the parent query, and set the SELECTED_COURSES query as the child query.

Map the A.COURSE child field to the A.COURSE parent field, and map the A.START_DATE child field to the A.START_DATE parent field.

Set the status of the connected query to *Active* and save the connected query.

Use Connected Query Viewer to view the course section schedule between *10/01/2009* and *12/31/09*.

Note. Use the *PTRPTG* for the user name and password in this activity.

Activity Detailed Steps

Perform the detailed steps to complete the activity.

Creating a Connected Query Using Connected Query Manager

To create a connected query using Connected Query Manager:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.
2. Select Reporting Tools, Connected Query, Connected Query Manager.
3. Click the Add a New Value link, and enter the following information:

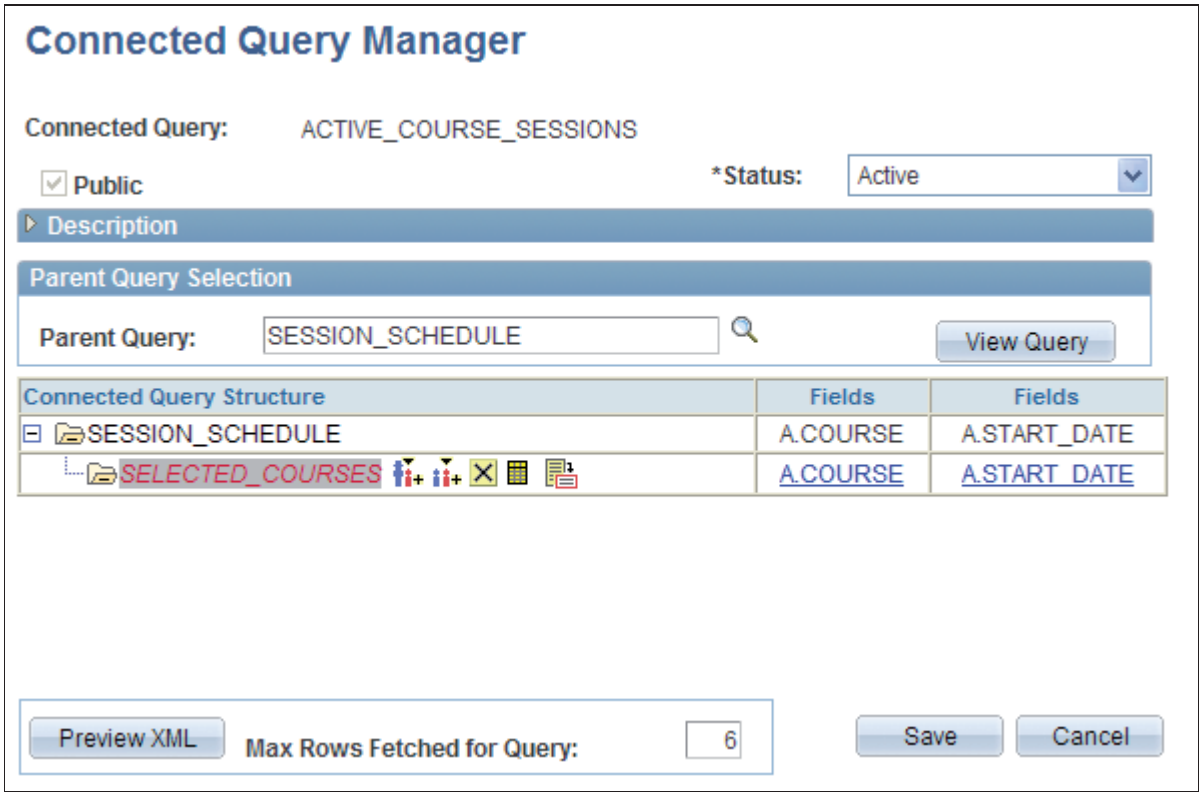
Page Element	***Value or Status***
Connected Query	*ACTIVE_COURSE_SESSIONS*
Public	Selected
Description	*Schedule of active sessions*

4. Use the Parent Query lookup icon to search and select the *SESSION_SCHEDULE* parent query.
5. In the Connected Query Structure section, click the Insert Child Query icon associated with the SESSION_SCHEDULE parent query.
6. Search and select the SELECTED_COURSES child query.
7. In the Connected Query Structure section, click the Select Related Fields icon associated with the SELECTED_COURSES child query.
8. Select the A.COURSE child field, and confirm that it is mapped with the A.COURSE parent field.
9. Select the A.START_DATE child query, and confirm that it is mapped with the A.START_DATE parent field.
10. Click the Map Fields button.
11. Select the *Active* option from the Status field, and save the connected query.
12. Click the Preview XML button.
13. View the section schedule from *10/01/2009* through *12/31/09.*

    Connected Query displays the XML report in a separate window.

Results

This example shows the Connected Query Manager displaying the ACTIVE_COURSE_SESSIONS connected query:



Viewing a Connected Query in Connected Query Viewer

To view a connected query in Connected Query Viewer:

1. Select Reporting Tools, Connected Query, Connected Query Viewer.
2. Search for the ACTIVE_COURSE_SESSIONS connected query, and click its Preview XML link.
3. View the section schedule from *10/01/2009* through *12/31/09*.

   Connected Query displays the XML report in a separate window.
4. Compare the outputs with the following results.

Results

This example shows the ACTIVE_COURSE_SESSIONS connected query in structured XML format:

```
<?xml version="1.0" ?>
- <ConnectedQuery name="ACTIVE_COURSE_SESSIONS" numrows="6" preview="True" query_rows_limit="6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="">
  - <SESSION_SCHEDULE>
    - <A.TRAINING_LOC>
        <![CDATA[ CORP ]]>
      </A.TRAINING_LOC>
      <A.START_DATE>12/29/2009</A.START_DATE>
      <A.END_DATE>12/29/2009</A.END_DATE>
    - <A.COURSE>
        <![CDATA[ 1016 ]]>
      </A.COURSE>
    - <A.INSTRUCTOR>
        <![CDATA[ RLD ]]>
      </A.INSTRUCTOR>
    - <SELECTED_COURSES>
      - <A.COURSE>
          <![CDATA[ 1016 ]]>
        </A.COURSE>
        <A.SESSION_NBR>201</A.SESSION_NBR>
      - <A.BUSINESS_UNIT>
          <![CDATA[ CORPS ]]>
        </A.BUSINESS_UNIT>
        <A.START_DATE>12/29/2009</A.START_DATE>
        <A.END_DATE>12/29/2009</A.END_DATE>
        <A.MAX_ENROLL>20</A.MAX_ENROLL>
      - <A.SESSION_STATUS>
          <![CDATA[ Active ]]>
        </A.SESSION_STATUS>
      - <A.INSTRUCTOR>
          <![CDATA[ RLD ]]>
        </A.INSTRUCTOR>
      - <A.TRAINING_LOC>
          <![CDATA[ CORP ]]>
        </A.TRAINING_LOC>
      - <A.CLASSROOM>
          <![CDATA[ Room B ]]>
        </A.CLASSROOM>
      </SELECTED_COURSES>
    </SESSION_SCHEDULE>
```

This concludes the activity. Please do not continue.

Review

In this lesson, you learned that:

- A connected query is a hierarchical object built with existing PeopleSoft queries. Connected Query enables you to create a single XML file based on a set of queries with parent-child relationships.
- The Connected Query Quick Start is a wizard that walks you through creating a simple connected query with a single parent query and a single child query linked using a set of related fields.
- Connected Query Manager enables a report developer to create, edit, copy, delete, and run connected queries.
- Connected Query Viewer is a read-only version of Connected Query Manager. It enables query users to view results of running connected queries using these three methods: Preview XML, Run to Window, and Schedule.
- You can schedule a connected query using the Run To Window link, the Schedule link, or accessing the Connected Query Scheduler component.

Slide 219

Student Notes

Additional Resources

This table lists additional resources that provide more details about the topics that we discussed in this lesson:

Topic	***Cross-Reference***
Understanding Connected Query	*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query,* "Using Connected Query"

Lesson 17

Course Workshop

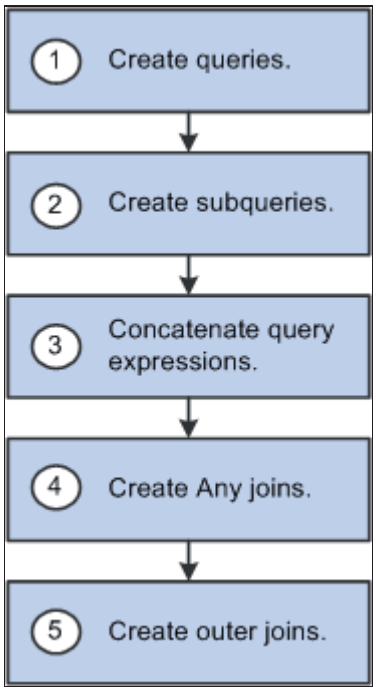
Objectives

By the end of this lesson, you will be able to apply the knowledge and skills that you learned throughout this course to create three queries using PeopleSoft Query.

Slide 221

Creating Queries

This diagram illustrates the steps that you will complete in this lesson to create queries using PeopleSoft Query:



Slide 222

Course Workshop Overview

This lesson is designed to challenge. Try to complete it using only the overview information that is provided here. If you need more details, refer to the step-by-step instructions in Appendix A.

See Appendix A, "Course Workshop Solution."

Setting the Scenario

The training manager needs three queries.

The first query should list courses having more than 10 total sessions scheduled.

Display the course name, course type, and course duration in the output. Course information is in the course record (PSU_COURSE_TBL), and you can find the session count by tallying the sessions in the Course session record (PSU_CRS_SESSN). Order the results by length of course. Name the first query MULTIPLE_SESSIONS.

Second, the Training department is interviewing students for an instructor position and needs a query that lists students who graduated with a grade point average (GPA) above 3.5.

Display the student ID, GPA, combined degree and major, skill, and proficiency level fields from the student education (PSU_STUDENT_ED) and student experiences (PSU_STUDENT_EXP) records. The output should appear in order of student ID, and the headings should be understandable. Name the second query INTERVIEW.

Finally, the training manager wants a list of students and degrees.

Display the student name, curriculum emphasis, project role, degree, and major. Include students without college degrees. Student information is stored in the student (PSU_STUDENT_TBL) and student education (PSU_STUDENT_ED) records. Please make the column headings and fields easily understandable, and sort the output by student name. Name the third query DEGREE_STATUS.

Note. Use *PTRPTG* for the user name and password in this activity.

Lesson 18

Course Review

Objectives

In this course, you learned how to:

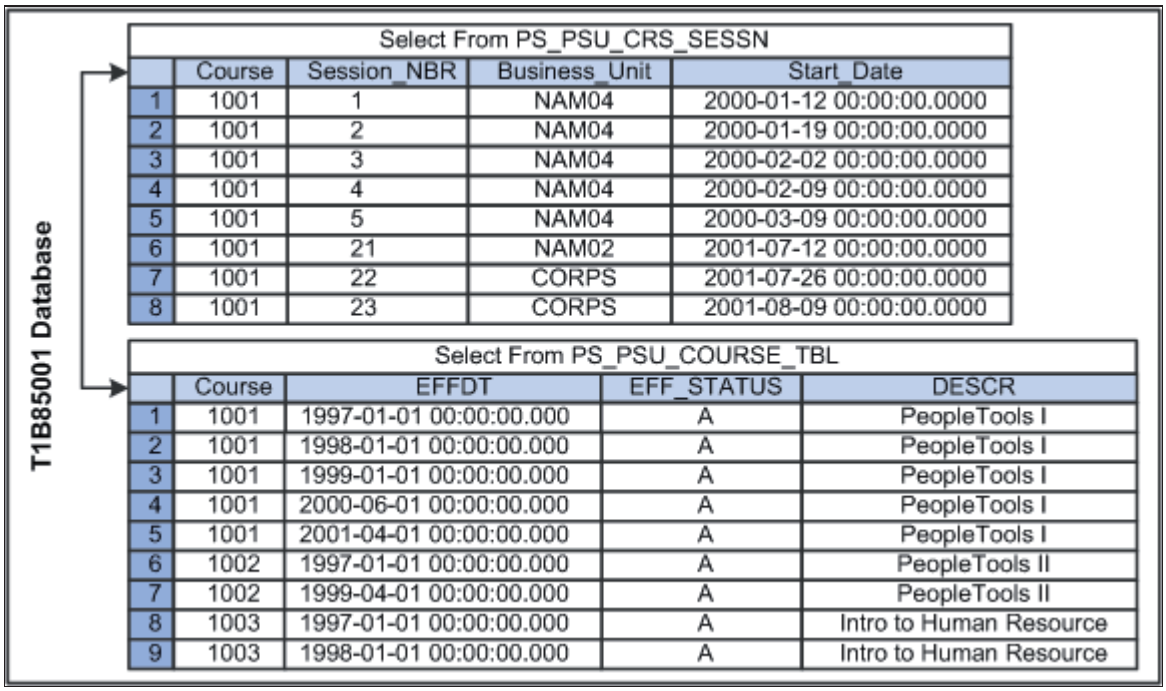
- Describe PeopleSoft Query reporting.
- Use Query Manager, and create a simple query.
- Filter outputs with criteria and runtime prompts.
- Use summary calculations, and join multiple tables to create queries.
- Perform administrative tasks.
- Create expressions.
- Use drilling URLs in PeopleSoft Query.
- Use the Any-join and out-join features.
- Use subqueries.
- Work with unions.
- Use Connected Query.

Slide 224

Describing PeopleSoft Query Basics

PeopleSoft Query

This diagram illustrates the tables (records), columns (fields), and rows (field data) that you find in database:



Select From PS_PSU_CRS_SESSN

	Course	Session_NBR	Business_Unit	Start_Date
1	1001	1	NAM04	2000-01-12 00:00:00.0000
2	1001	2	NAM04	2000-01-19 00:00:00.0000
3	1001	3	NAM04	2000-02-02 00:00:00.0000
4	1001	4	NAM04	2000-02-09 00:00:00.0000
5	1001	5	NAM04	2000-03-09 00:00:00.0000
6	1001	21	NAM02	2001-07-12 00:00:00.0000
7	1001	22	CORPS	2001-07-26 00:00:00.0000
8	1001	23	CORPS	2001-08-09 00:00:00.0000

Select From PS_PSU_COURSE_TBL

	Course	EFFDT	EFF_STATUS	DESCR
1	1001	1997-01-01 00:00:00.000	A	PeopleTools I
2	1001	1998-01-01 00:00:00.000	A	PeopleTools I
3	1001	1999-01-01 00:00:00.000	A	PeopleTools I
4	1001	2000-06-01 00:00:00.000	A	PeopleTools I
5	1001	2001-04-01 00:00:00.000	A	PeopleTools I
6	1002	1997-01-01 00:00:00.000	A	PeopleTools II
7	1002	1999-04-01 00:00:00.000	A	PeopleTools II
8	1003	1997-01-01 00:00:00.000	A	Intro to Human Resource
9	1003	1998-01-01 00:00:00.000	A	Intro to Human Resource

Slide 225

Student Notes

PeopleSoft Query and Databases

PeopleSoft Query is an end-user reporting tool. You use queries to retrieve information from the database to a user interfaces.

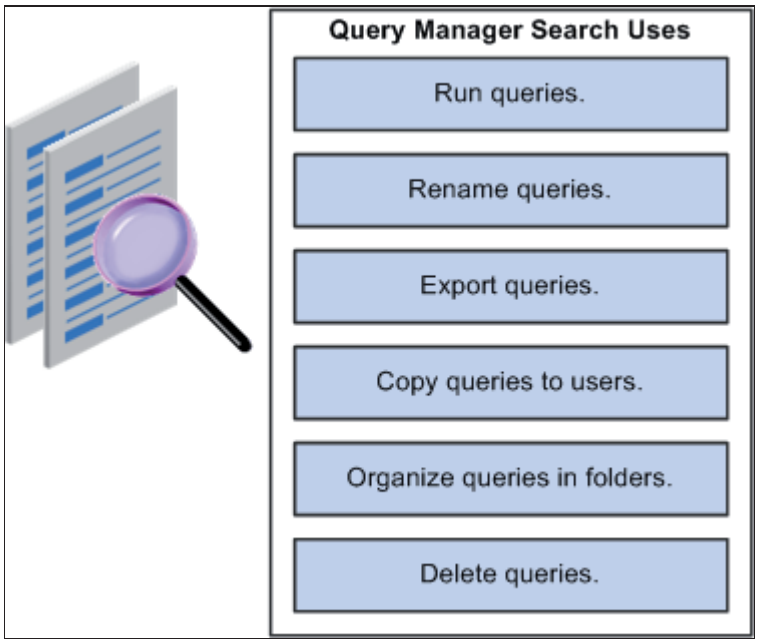
Database elements include:

- Tables (records)
- Columns (fields)
- Rows (field data)
- Keys

Using Query Manager

Query Manager Search

This illustration shows the usages of the Query Manager search page:



Slide 226

Student Notes

Query Manager

Query Manager:

- Is a browser-based tool.
- Is organized into nine pages that make creating queries more intuitive.
- Assists you in creating queries without you having to know SQL.

Creating a Simple Query

Methods to Create a Query

You can use either of these two methods to create a query:

1. Click the Create New Query link on the Query Manager search page.
2. Click the New Querylink at the bottom of the pages that are in the Query Manager component.

Slide 227

Student Notes

Creating a Query

To create a query:

1. Access Query Manager.
2. Select a record and the fields.
3. Edit field properties by changing field headings and applying aggregate functionality to fields.
4. Access the Query Properties page to provide additional information to the query.
5. Remove duplicate rows by using the Distinct feature.
6. Save and run the query.

Filtering Output by Using Criteria

Purpose of Applying Criteria to Queries

You define criteria rows in the query to:

- Reduce the number of rows of data that are returned.
- Retrieve only the information that you need at the time that the query runs.

Slide 228

Student Notes

Condition Types

Query Manager provides 18 conditions that you can apply to the criteria.

The commonly used conditions in query criteria are:

- Equal to
- Like
- Is null
- Between
- In tree

Boolean Expressions

These are four types of Boolean expressions that are available from the Criteria page:

- AND: Returns rows of data if criteria are true.
- AND NOT: Does not return rows of data if the criteria are true.
- OR: Returns rows of data if any of the rows in the criteria are true.
- OR NOT: Does not return rows of data if any of the rows in the criteria are true.

Effective Date

The effective date (EFFDT) field is used throughout PeopleSoft applications to enable the viewing of data that changes over time.

You can use effective-dated fields in criteria to:

- View all rows, regardless of their effective dates.

- View rows that aren't currently in effect.
- View the rows that were effective as of a particular date.

Filtering Output by Runtime Prompts

Runtime Prompts

You use prompts to give more control and flexibility over selection criteria values at runtime. When prompted, you enter a value for a specific field at the time that the query is run.

This diagram lists the four types of edits available for runtime prompts:

Prompt Type	Source of Value
1 Prompt Table	User selects from a list of values defined in another application data table.
2 Translate Table	User selects from a list of values defined in a translate table.
3 Yes/No Table	User selects either yes or no.
4 No Table Edit	User enters a value that does not require an edit.

Slide 229

Student Notes

Runtime Prompts

Runtime prompts enable user input into how Query Manager filters data at the time that the query is run.

Create runtime prompts from the Prompts page or the Criteria page in Query Manager.

You add one or multiple runtime prompts, depending on the amount of user input needed to filter the data.

Using Summary Calculations

Aggregate Functions

You can associate fields in a query with basic mathematical calculations called aggregate functions.

In Query Manager, aggregate functions:

- Enable you to associate query fields with predefined calculations.
- Return a single value for multiple rows of output.

Slide 230

Student Notes

Aggregate Functions

When you apply an aggregate function to a field, PeopleSoft Query displays the results of the function in the field, wherever it occurs.

The aggregate functions in Query Manager are:

- Avg
- Count
- Max
- Min
- Sum

Having Criteria

PeopleSoft Query provides the Having page to enable you to add criteria on the aggregate not on the field generating the aggregate.

The Having criteria appear in the HAVING clause of SQL statement.

Joining Multiple Tables to Create Queries

Joining Tables

When you join two records (tables), you relate them to each other.

You join two records to:

- Retrieve additional fields.
- Limit the rows that are returned.

Slide 231

Student Notes

Predefined Joins

In PeopleSoft Query, a predefined join is one of the following:

- Record-hierarchy join.

  A record-hierarchy join uses records that are related through a parent-child relationship. You create this relationship when you define record properties in Application Designer.

- Related-record join.

  Related-record joins combine nonhierarchical records that are related by common fields. You create this relationship when you define prompt table properties in Application Designer.

Performing Administrative Tasks

Administrative Tools

Use Query Viewer when users need only to view a query report but do not need capabilities such as deleting, renaming, creating, or modifying queries.

Use Schedule Query to run queries at specified dates and times.

Use the Query Administration component to monitor query performance and use.

Slide 232

Student Notes

Query Administration Component

You use the Query Administration component to:

- Keep logs on queries.
- Monitor average runtime.
- Monitor the number of running times.
- Monitor last running date and time.
- Retrieve queries based on runtime length.
- Terminate queries.
- Assign new owners to queries.

Creating Expressions

Query Expressions

Query expressions are calculations that Query Manager performs as part of a query.

You use query expressions to calculate a value that PeopleSoft Query does not provide by default; for example, to add values from two fields together or to multiply a field value by a constant.

Slide 233

Student Notes

Two Methods to Use Expressions

You use expressions:

- As column output.
- In criteria to filter data.

Expressions as Columns or Fields

When you create expressions, you might use them as columns or fields.

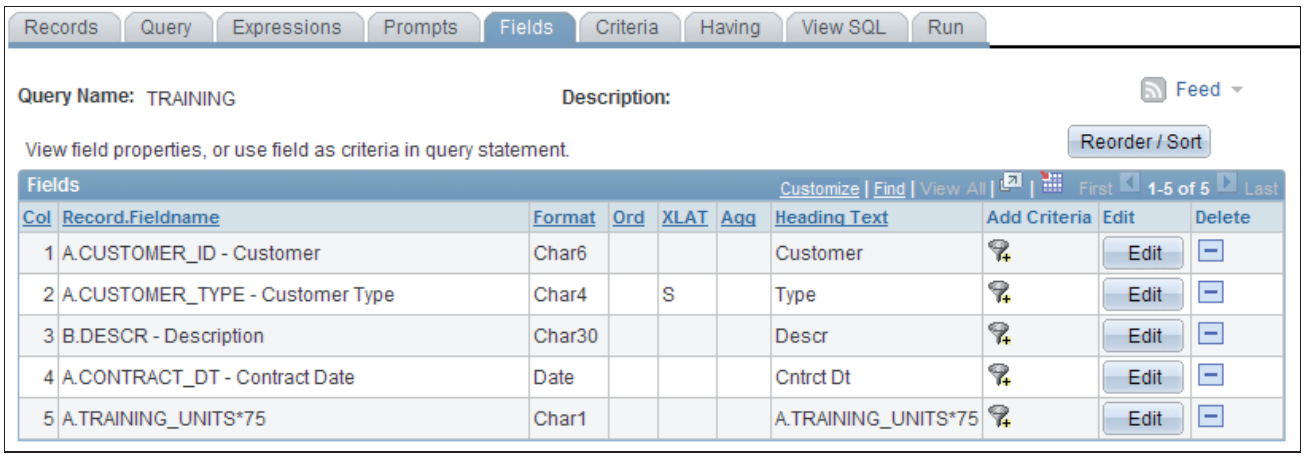
This example uses an expression as a column:

	Customer	Type	Units	Total Cost of Units *125
As columns in the query output.	ABN	FRND	400	50000
	ALBRAS	FRND	1000	125000

	Customer	Type	Units	(Total Cost of Units *125) > 60000
In criteria to filter out data.	ALBRAS	FRND	1000	125000

When you use an expression as a field, the expression functions as any other field you use in a query.

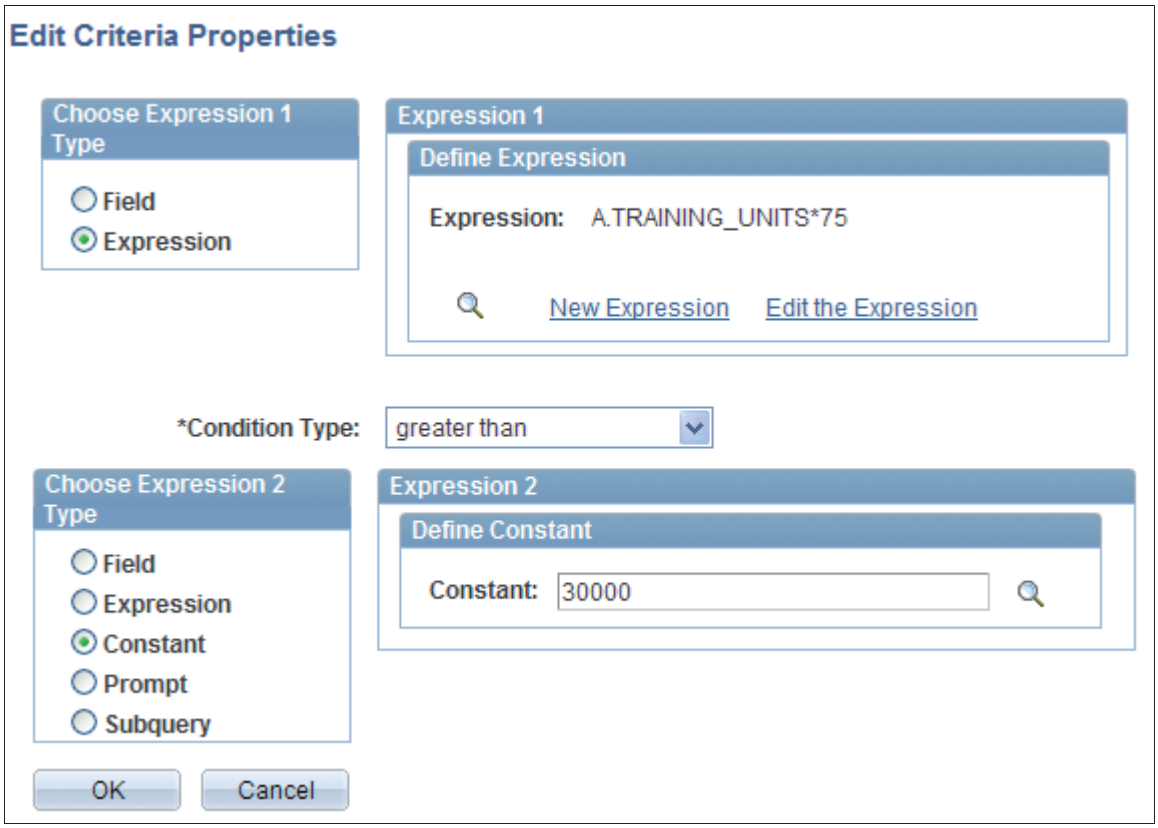
When you preview the query, the expression name appears as the column heading in the query, as in this example:



Expressions Used in Criteria

Use expressions in criteria to filter data. For instance, an expression multiplies customers' total training units by 75.

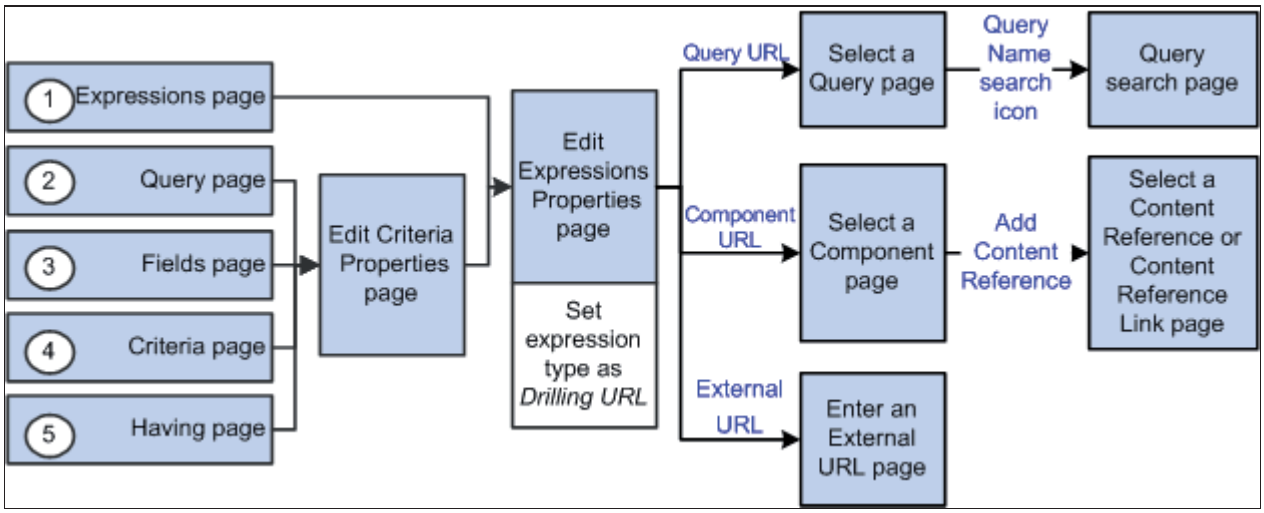
You apply the expression to a row of criteria to return rows that are greater than 30,000 training units, as this example shows:



Using Drilling URLs in PeopleSoft Query

Drilling URLs

This diagram shows a high-level overview of how to build drilling URLs:



Slide 234

Student Notes

Drilling URLs

Drilling URLs are the URLs that you define by selecting the menu, component, page, portal object, or URL of choice.

When you build a query using Query Manager, you can define drilling URLs that are associated with this query. These settings are saved into the database, along with prompt, criteria, and so on, as part of the metadata for this query. When you execute this query through Query Manager or Query Viewer, the query results page shows results as links, which you can click to be redirected to a different page in a new browser.

Depending on how drilling URLs are defined, the new browser is either a PeopleSoft Pure Internet Architecture page, another query result page, or an external page.

Implementing the Any-Join Feature

Any Joins

You create queries that access multiple tables, even when the tables are not in the parent hierarchy or the related-record hierarchy.

You manually link the tables to retrieve the correct output. You link the tables using key fields.

Slide 235

Student Notes

Example: Join Tables with Any Joins

Suppose that you have the records and fields as shown in this table:

Record	*Field*
PSU_STUDENT_TBL	STUDENT_ID (Key field) NAME CUSTOMER_ID
PSU_CUST_TBL	CUSTOMER_ID (Key field) DESCR

The WHERE clause uses the CUSTOMER_ID field to join these two records

```
SELECT A.STUDENT_ID,
         A.NAME,
         B.DESCR
FROM  PSU_STUDENT_TBL A,
         PSU_CUST_TBL B
WHERE  A.CUSTOMER_ID = B.CUSTOMER_ID
```

Using Subqueries

Subqueries

A subquery is a query within a query that you use to compare the value for a field in the subsuming query to the results of a second query.

You reference the subquery in the WHERE clause, which you access on the Criteria page in Query Manager.

Use subqueries to:

- Produce a single value for comparison.

  Condition type: equal to (aggregate value).

- Return a value of true or false.

  Condition type: exist / does not exist.

- Produce a list of values.

  Condition type: in list / not in list.

Slide 236

Student Notes

Subqueries

Subqueries enable you to compare the value for a field in the current query to the results of a second query.

Note the following information about subqueries:

- The condition type that you specify in the criteria determines what the subquery returns to the query.
- A subquery can join two tables, but the subquery can retrieve only one data value.

  You use this feature to specify criteria that is based on two records.

- The result of the subquery itself does not appear; the subquery *limits* the data that the query retrieves.
- Additional rows of criteria can be placed in the parent query or the subquery.
- To set up a subquery, access the Criteria page, select Subquery as the expression type for expression 2, and then click the Define/Edit Subquery link.

Working with Unions

Unions

You use unions to combine records that have no fields in common and to retrieve similar data from unrelated records in one query.

A union combines two SELECT statements in the same query.

Slide 237

Student Notes

Union Rules to Follow

Unions enable you to have two tables in the same query without having joining criteria and without creating a Cartesian product.

Follow three rules with unions. Each SELECT statement must include:

- The same number of fields.
- Corresponding field types.
- The same field order for each select statement.

Performing Outer Joins

Outer Joins

In an outer join, all rows of data are included from the master table. Matching rows from the subordinate table are also included.

With a *left outer join,* all rows from table 1 [for example, the PSU Courses table (PSU_COURSE_TBL)] appear in the result, even if no match exists in table 2 [for example, no course IDs are in the PSU Course Session table (PSU_CRS_SESSN)].

Slide 238

Student Notes

Purpose of Outer Joins

Outer joins combine aspects of record-hierarchy joins and subqueries.

Remember that a record-hierarchy join retrieves rows for which fields match from different tables, for example, A.Field1 equals B.Field1.

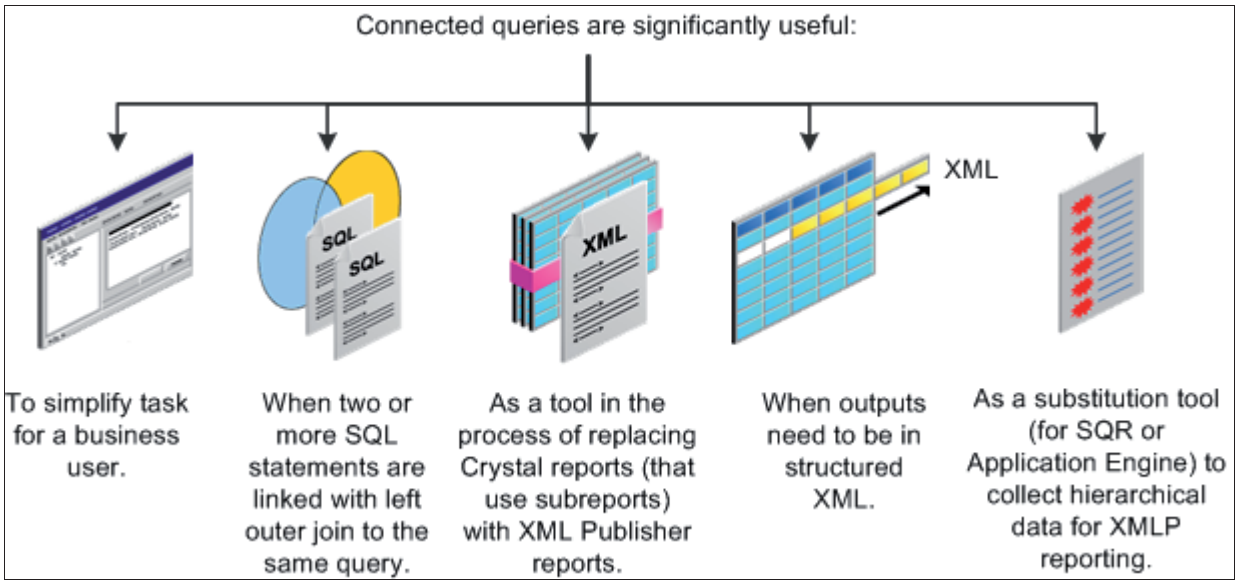
A subquery retrieves rows from the primary table that do not have matches in the secondary table.

Using Connected Query

Connected Query

Connected Query enables you to create a single XML file based on a set of queries with parent-child relationships.

This diagram shows the usages of connected queries:



Slide 239

Student Notes

Connected Query

Notice that:

- A connected query is a hierarchical object built with existing PeopleSoft queries.

  Connected Query enables you to create a single XML file based on a set of queries with parent-child relationships.

- The Connected Query Quick Start is a wizard that walks you through creating a simple connected query with a single parent query and a single child query linked using a set of related fields.

- Connected Query Manager enables a report developer to create a new connected query, edit an existing connected query, copy existing connected queries, and delete a connected query object, as well as run connected queries.

- Connected Query Viewer is a read-only version of Connected Query Manager.

  It enables query users to view results of running connected queries using these three methods: Preview XML, Run to Window, and Schedule.

- You can schedule a connected query in three ways using the Run To Window link, the Schedule link, or accessing the Connected Query Scheduler component.

Appendix A

Course Workshop Solution

Setting the Scenario

The training manager needs three queries.

The first query should list courses having more than 10 total sessions scheduled.

Display the course name, course type, and course duration in the output. Course information is in the course record (PSU_COURSE_TBL), and you can find the session count by tallying the sessions in the Course session record (PSU_CRS_SESSN). Order the results by length of course. Name the first query MULTIPLE_SESSIONS.

Second, the Training department is interviewing students for an instructor position and needs a query that lists students who graduated with a grade point average (GPA) above 3.5.

Display the student ID, GPA, combined degree and major, skill, and proficiency level fields from the student education (PSU_STUDENT_ED) and student experiences (PSU_STUDENT_EXP) records. The output should appear in order of student ID, and the headings should be understandable. Name the second query INTERVIEW.

Finally, the training manager wants a list of students and degrees.

Display the student name, curriculum emphasis, project role, degree, and major. Include students without college degrees. Student information is stored in the student (PSU_STUDENT_TBL) and student education (PSU_STUDENT_ED) records. Please make the column headings and fields easily understandable, and sort the output by student name. Name the third query DEGREE_STATUS.

Note. Use *PTRPTG* for the user name and password in this activity.

Creating Query

To create the base query:

1. If necessary, sign in to the Oracle PeopleSoft Enterprise.

2. Access Query Manager, and create a new query using the following information:

Record	Field (Property)
PSU_COURSE_TBL	DESCR *(Course Name)* COURSE_TYPE LENGTH_DAYS *(RFT Long,* Order by 1)

3. Save the query as MULTIPLE_SESSIONS.
4. Access the Criteria page, and add a row of criteria using the following information:

Page Element	Value or Status
Expression 1 Type	*Field*
Expression 1	<blank>
Condition type	*exists*
Expression 2 type	*Subquery*

5. Click the Define/Edit Subquery link for Expression 2.

Creating Subqueries

To create subqueries:

1. On the Records page, search for and add the PSU_CRS_SESSN record.
2. Select the SESSION_NBR field.
3. Select the *Count* aggregate option for the SESSION_NBR field.
4. Access the Having page, and add a Having criteria using the following information:

Page Element	Value or Status
Expression 1 Type	*Field*
Expression 1	*B.SESSION_NBR*
Condition Type	*greater than*
Expression 2 Type	*Constant*
Expression 2	*10*

5. On the Criteria page, add a criteria using the following information:

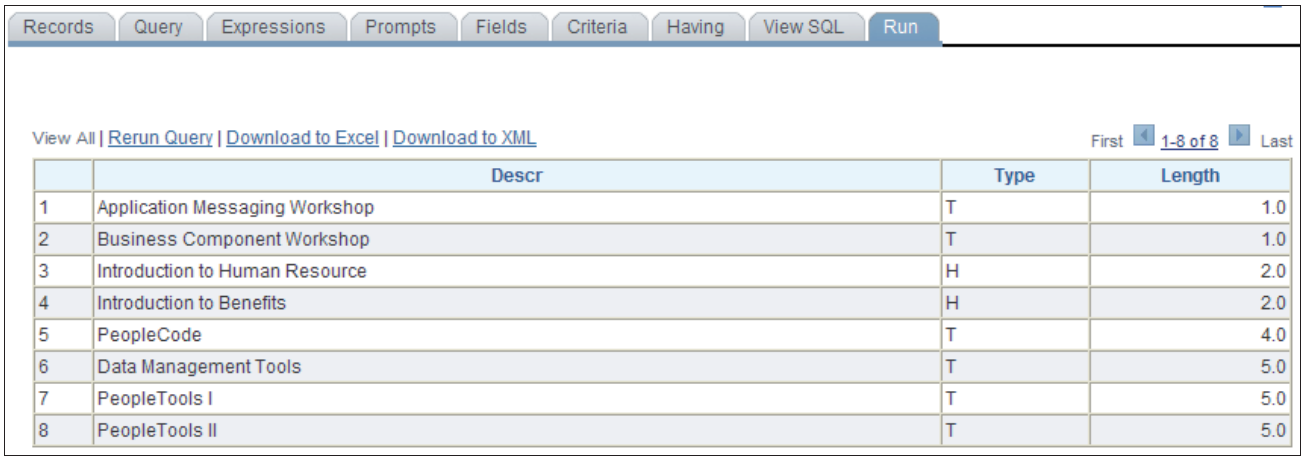
Page Element	***Value or Status***
Expression 1 Type	*Field*
Expression 1	*B.COURSE*
Condition Type	*equal to*
Expression 2 Type	*Field*
Expression 2	*A.COURSE*

6. Save and preview the MULTIPLE_SESSIONS query.

7. Compare the output with the following results.

Results

The MULTIPLE_SESSIONS query returns eight rows:



	Descr	Type	Length
1	Application Messaging Workshop	T	1.0
2	Business Component Workshop	T	1.0
3	Introduction to Human Resource	H	2.0
4	Introduction to Benefits	H	2.0
5	PeopleCode	T	4.0
6	Data Management Tools	T	5.0
7	PeopleTools I	T	5.0
8	PeopleTools II	T	5.0

Concatenating Query Expressions

To concatenate query expressions:

1. Create a new query base on these specifications:

Record	***Field (Property)***
PSU_STUDENT_ED	STUDENT_ID *(RFT Long,* Order by 1) GPA (*RFT Short)*

2. Access the Criteria page, and add a new criteria using the following information:

Page Element	***Value or Status***
Expression 1 type	*Field*
Expression 1	*A.GPA*
Condition type	*greater than*
Expression 2 type	*Constant*
Expression 2	*3.5*

3. Save the query as INTERVIEW.
4. Access the Expressions page, and add an expression using the following information:

Page Element	***Value or Status***
Expression Type	*Character*
Length	*30*
Expression Text	*A.DEGREE %CONCAT + ' - ' %CONCAT + A.MAJOR*

5. Click the OK button, and click the Use as Field link.
6. Edit the heading text of the expression field as *Degree - Major*.
7. Save the INTERVIEW query.

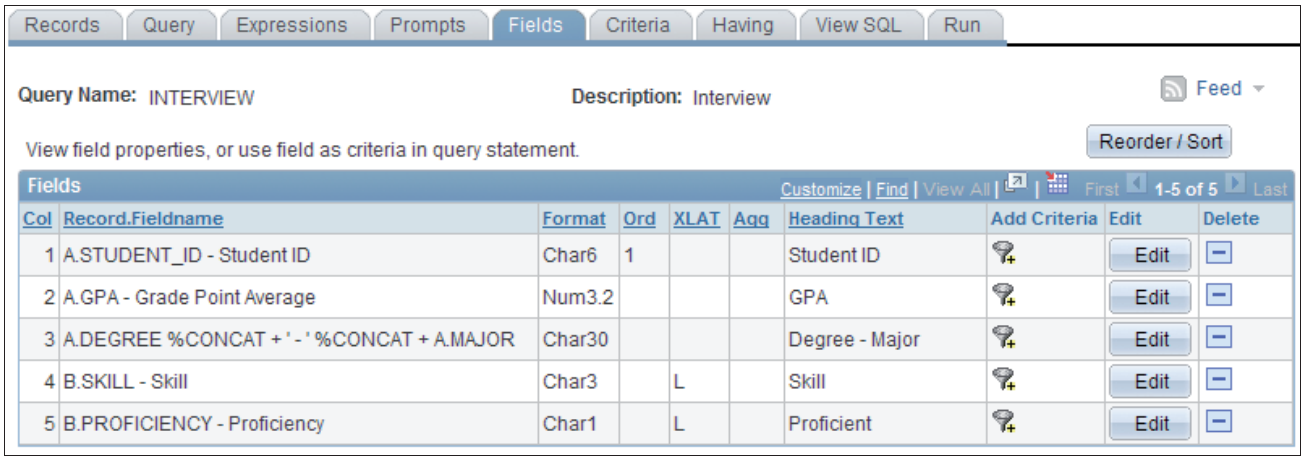
Creating Any Joins

To create any joins:

1. On the Records page, search for and join the PSU_STUDENT_EXP record.
2. Accept the standard join option, and click the A=PSU_STUDENT_ED link.
3. Click the Add Criteria button.
4. Select the SKILL (XLAT Long) and PROFICIENCY (XLAT Long) fields.
5. Save and preview the query.
6. Compare the output with the following results.

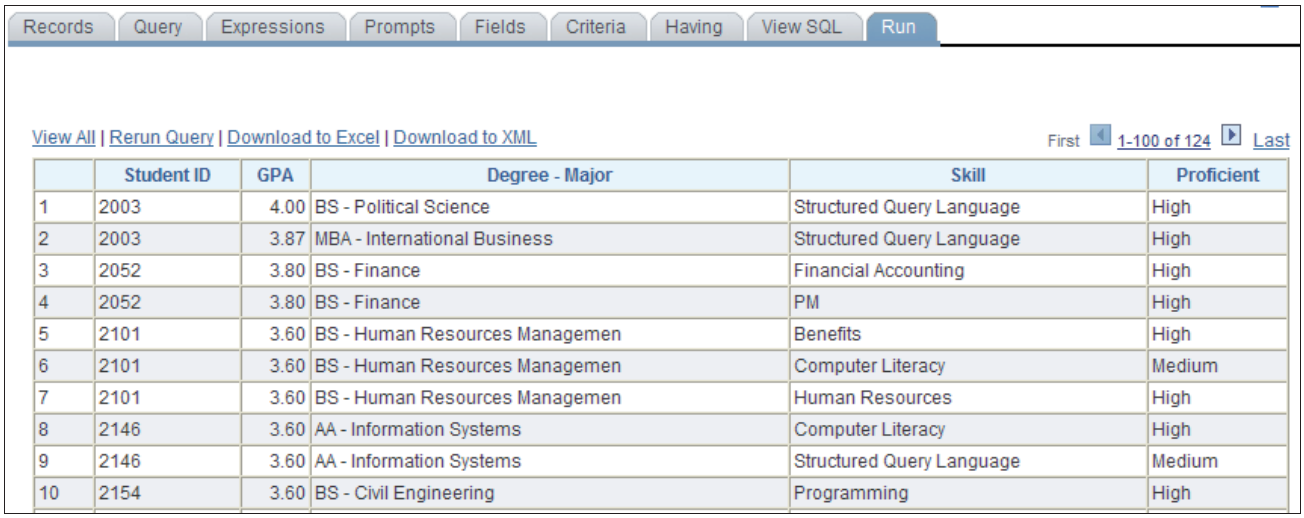
Results

This example shows the Fields page after you concatenated query expressions and created any-join:



Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete
1	A.STUDENT_ID - Student ID	Char6	1			Student ID		Edit	
2	A.GPA - Grade Point Average	Num3.2				GPA		Edit	
3	A.DEGREE %CONCAT + ' - ' %CONCAT + A.MAJOR	Char30				Degree - Major		Edit	
4	B.SKILL - Skill	Char3		L		Skill		Edit	
5	B.PROFICIENCY - Proficiency	Char1		L		Proficient		Edit	

The INTERVIEW query returns 124 rows:



	Student ID	GPA	Degree - Major	Skill	Proficient
1	2003	4.00	BS - Political Science	Structured Query Language	High
2	2003	3.87	MBA - International Business	Structured Query Language	High
3	2052	3.80	BS - Finance	Financial Accounting	High
4	2052	3.80	BS - Finance	PM	High
5	2101	3.60	BS - Human Resources Managemen	Benefits	High
6	2101	3.60	BS - Human Resources Managemen	Computer Literacy	Medium
7	2101	3.60	BS - Human Resources Managemen	Human Resources	High
8	2146	3.60	AA - Information Systems	Computer Literacy	High
9	2146	3.60	AA - Information Systems	Structured Query Language	Medium
10	2154	3.60	BS - Civil Engineering	Programming	High

Creating Outer Joins

To create outer joins:

1. If necessary, sign in to PeopleSoft PeopleTools 8.50.
2. Create a new query base on these specifications:

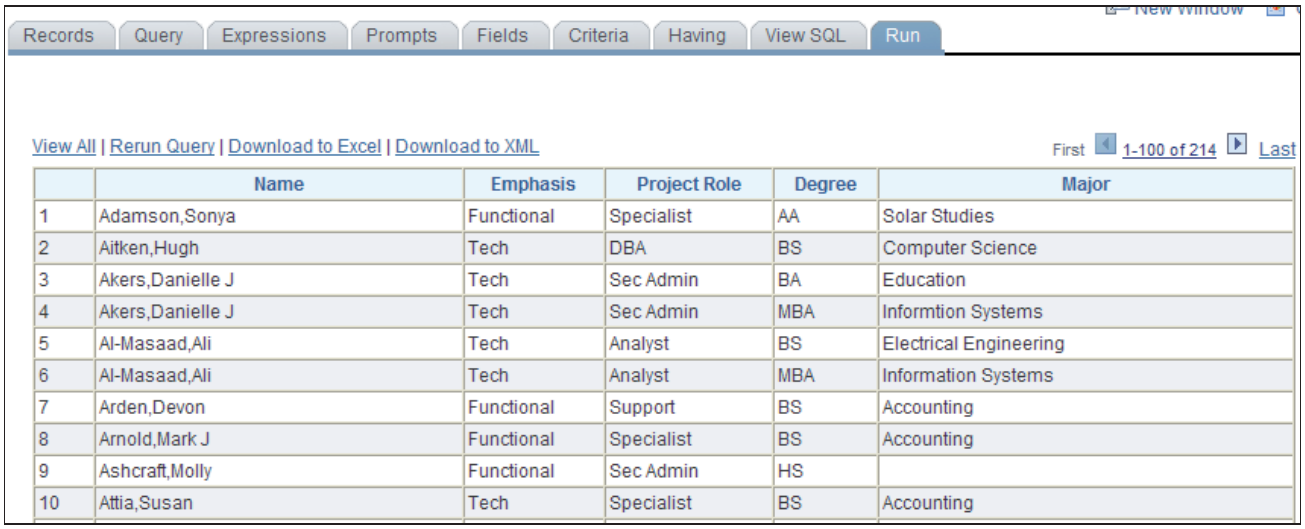
Record	***Field (Property)***
PSU_STUDENT_TBL	STUDENT_NAME (Order by 1) EMPHASIS (XLAT Short) PROJECT_ROLE (XLAT Short)

3. Save the query as DEGREE_STATUS.
4. Access the Records page.
5. Search for and join the PSU_STUDENT_ED record.

6. Accept the standard join option, and click the A=PSU_STUDENT_TBL link.
7. Click the Add Criteria button.
8. Select the DEGREE (XLAT Short) and MAJOR fields.
9. Save and preview the query.
10. Compare the output with the following results.

Results

The DEGREE_STATUS query returns 214 rows:



	Name	Emphasis	Project Role	Degree	Major
1	Adamson,Sonya	Functional	Specialist	AA	Solar Studies
2	Aitken,Hugh	Tech	DBA	BS	Computer Science
3	Akers,Danielle J	Tech	Sec Admin	BA	Education
4	Akers,Danielle J	Tech	Sec Admin	MBA	Informtion Systems
5	Al-Masaad,Ali	Tech	Analyst	BS	Electrical Engineering
6	Al-Masaad,Ali	Tech	Analyst	MBA	Information Systems
7	Arden,Devon	Functional	Support	BS	Accounting
8	Arnold,Mark J	Functional	Specialist	BS	Accounting
9	Ashcraft,Molly	Functional	Sec Admin	HS	
10	Attia,Susan	Tech	Specialist	BS	Accounting

Appendix B

Defining Record-Hierarchy and Related-Record Joins

Join Types

You use joins to retrieve data from multiple records or to specify criteria from multiple records.

Whenever you perform a join, the records are linked based on common fields.

PeopleSoft Application Designer enables you to configure predefined joins so that you do not have to add criteria that links the records (view the SQL for more details.)

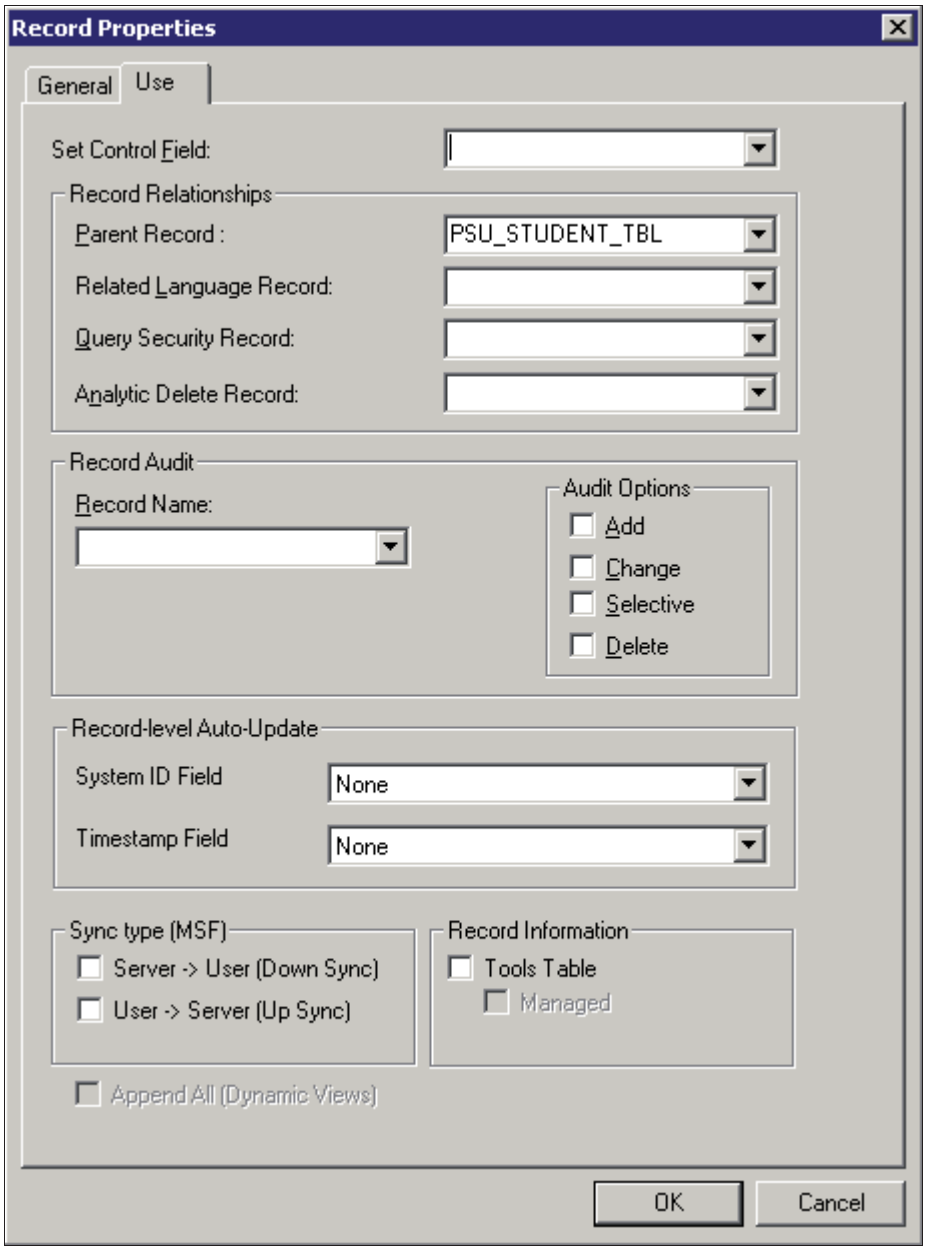
Application Designer enables you to configure these predefined joins:

- Record-hierarchy join
- Related-record join

Record-Hierarchy Join

A hierarchical join uses records that are related through a parent-child relationship. The join is defined by record properties and key structure when the record is created in PeopleSoft Application Designer.

For instance, the PSU Student Education table (PSU_STUDENT_ED) is a child record of the PSU Student Data table (PSU_STUDENT_TBL.) This record hierarchy was set in the record properties of the PSU_STUDENT_ED record definition, as shown in the following example:

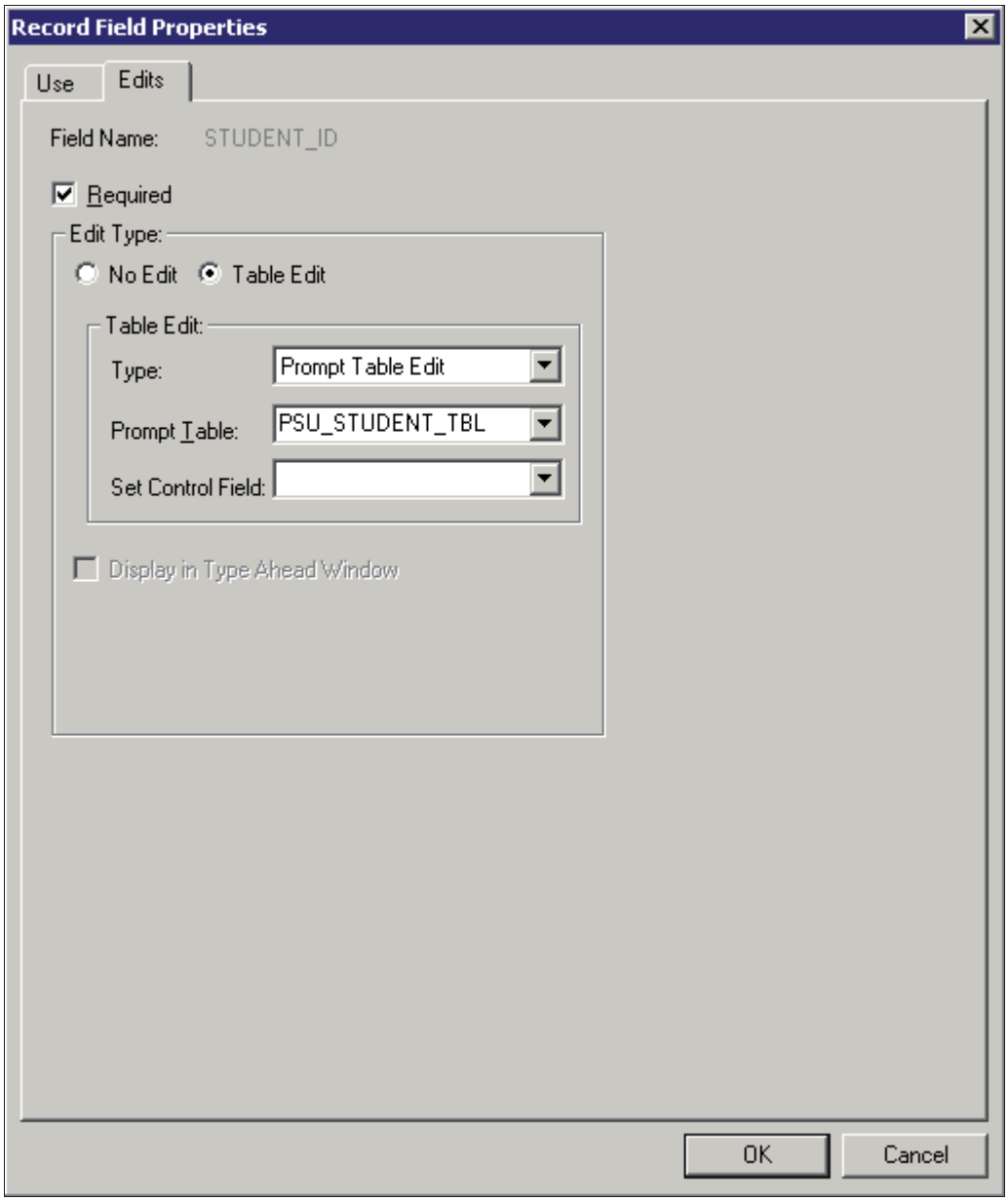


Related-Record Joins

Related-record joins use records from nonhierarchical records that are related by common fields. This is determined by the prompt table relationship defined for a field in PeopleSoft Application Designer.

For instance, the PSU_STUDENT_ED record definition has six fields, one of which is the STUDENT_ID field. The STUDENT_ID field prompts for values from the PSU_STUDENT_TBL table.

The prompt table (related record) property is set in the record field properties of the PSU_STUDENT_ED record definition, as shown in the following example:



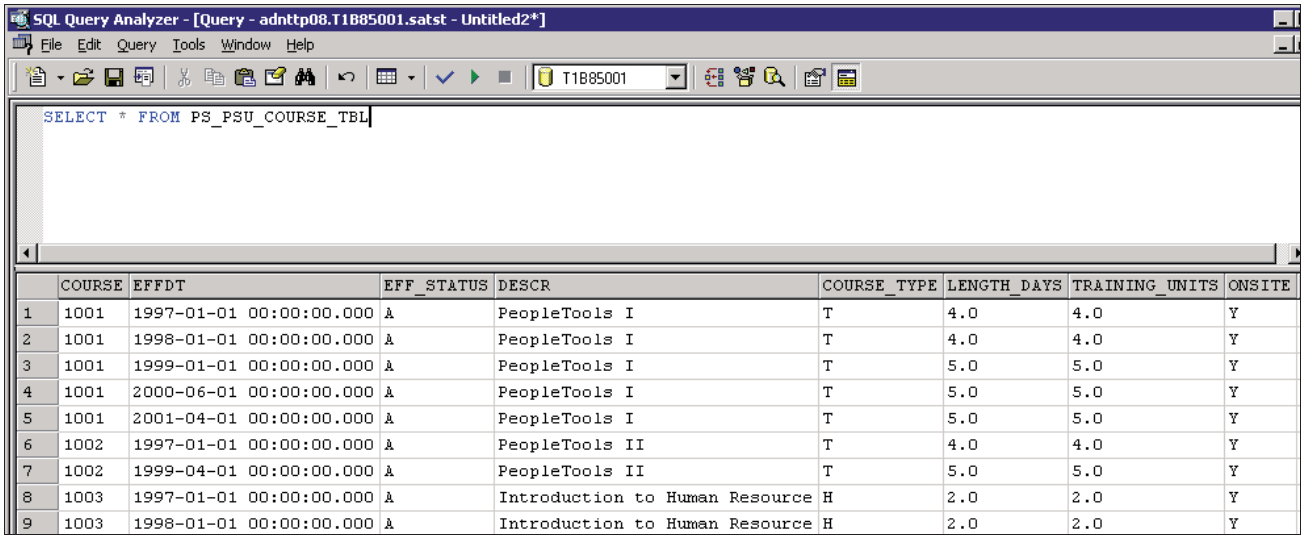
Appendix C

Writing Queries with SQL

Using Query Analyzer

Query Analyzer is a Microsoft SQL Server tool that enables you to enter SQL statements to retrieve data from the database.

Use Query Analyzer to retrieve information from a selected database:



	COURSE	EFFDT	EFF_STATUS	DESCR	COURSE_TYPE	LENGTH_DAYS	TRAINING_UNITS	ONSITE
1	1001	1997-01-01 00:00:00.000	A	PeopleTools I	T	4.0	4.0	Y
2	1001	1998-01-01 00:00:00.000	A	PeopleTools I	T	4.0	4.0	Y
3	1001	1999-01-01 00:00:00.000	A	PeopleTools I	T	5.0	5.0	Y
4	1001	2000-06-01 00:00:00.000	A	PeopleTools I	T	5.0	5.0	Y
5	1001	2001-04-01 00:00:00.000	A	PeopleTools I	T	5.0	5.0	Y
6	1002	1997-01-01 00:00:00.000	A	PeopleTools II	T	4.0	4.0	Y
7	1002	1999-04-01 00:00:00.000	A	PeopleTools II	T	5.0	5.0	Y
8	1003	1997-01-01 00:00:00.000	A	Introduction to Human Resource	H	2.0	2.0	Y
9	1003	1998-01-01 00:00:00.000	A	Introduction to Human Resource	H	2.0	2.0	Y

Query Analyzer Commands

Use these methods to run Query Analyzer commands:

- Press Ctrl+E.
- Press F5.
- Click the Execute toolbar button.

Query Analyzer Rules

When you use Query Analyzer:

- Enter table and column names in upper case.

- Separate column names with commas.
- Enclose alphanumeric constant values in single quotation marks.
- Maintain proper case within single-quoted strings (SQL is case-sensitive).
- Press Ctrl + End to view the number of rows that are returned.
- Enter queries in the Query section.
- View results in the Result section.

Using Basic SQL Commands

Use the basic SQL commands to retrieve information from the database.

SQL SELECT Statement

The SELECT statement is the key SQL statement that is used by PeopleSoft Query, PS/nVision, and Structured Query Report (SQR).

The SELECT command finds, retrieves, and displays data from tables or views.

This table lists the basic syntax of the SELECT statement:

Syntax	***Required/Optional***
SELECT <column or expression>	Required
FROM <tables and/or views>	Required
WHERE <column><relational predicate/operator><value>	Optional, except required for table joins. Use to place criteria on the select statement.
GROUP BY <columns>	Optional. Use to select fields and aggregates together.
ORDER BY <columns>	Optional. Use to sort output results.

SQL SELECT Command Expressions

The SELECT command uses the following expressions:

Expression	***Example***
Column names	EFFDT
Constants	Text
System keywords	getdate()
Results of functions	sum(rate)
Wildcards	*

SQL Wildcard Characters

The asterisk is a wildcard character for all columns in the table or view.

This is an example of the SQL wildcard:

```
SELECT * FROM  <table or view>
```

Some query languages support the system keyword ALL in place of the asterisk.

Example: SQL Wildcard

This statement retrieves all of the data in the Course table (PSU_COURSE_TBL):

```
SELECT * FROM PS_PSU_COURSE_TBL
```

Additional SQL Wildcards

When you are using the LIKE predicate to search for strings that match a specified pattern, the underscore (_) and the percent (%) are the pattern-matching wildcard characters in Query Analyzer.

This example shows the underscore that matches any single character:

```
WHERE COURSE like ' _1'
```

This example shows the percent sign that matches any sequence of zero or more characters:

```
WHERE COURSE like ' 1%'
```

Removing Duplicate Data

The Distinct command suppresses duplicate rows of output.

This command retrieves a single row for each employee:

```
SELECT DISTINCT COURSE, DESCR FROM PS_PSU_COURSE_TBL
```

SQL WHERE Clause

The WHERE clause in the SELECT statement specifies a search condition for the rows of data; those are contained in the table or view.

Use this syntax for the WHERE clause:

```
WHERE <column> <relational predicate> <value>
```

SQL ORDER BY Clause

Use the ORDER BY clause to specify the row order in the result.

You can use multiple columns to order rows.

The first column in the ORDER BY clause sets the highest level of the sort, followed sequentially by the other columns in the clause.

Example: Using the ORDER BY Clause

Use this SQL statement to produce an active course session listing for all course codes that begin with the number 1 ordered by date:

```
SELECT   COURSE,
            SESSION_NBR,
            START_DATE
FROM   PS_PSU_CRS_SESSN
WHERE   SESSION_STATUS = 'A'
            AND COURSE LIKE '1%'
ORDER BY  START_DATE,
            COURSE
```

Enter the DESC command after a column name to display the rows in descending order:

```
SELECT   COURSE,
            SESSION_NBR,
            START_DATE
FROM   PS_PSU_CRS_SESSN
WHERE   SESSION_STATUS = 'A'
            AND COURSE LIKE '1%'
ORDER BY  START_DATE DESC,
            COURSE
```

Note. The default order is ascending.

Adding Aggregate Functions in SQL Statements

Aggregate functions calculate one summary value from a group of values.

Available functions and the syntax vary according to the query tool.

Check the query tool to find the equivalent syntax for each database platform.

SQL Predefined Functions

Most query languages provide predefined functions for manipulating numbers, strings, and dates.

This table lists some aggregate functions and examples:

Function	*Example*
Avg	Select AVG(LENGTH_DAYS) From PS_PSU_COURSE_TBL
Count	Select COUNT(*) From PS_PSU_STUDENT_TBL
Max	Select MAX(START_DT) From PS_PSU_CRS_SESSN
Min	Select MIN(END_DATE) From PS_PSU_CRS_SESSN
Sum	Select SUM(TRAINING_UNITS) From PS_PSU_COURSE_TBL

Joining Tables in a SQL Query

When you need data from more than one table or view, the query must join the tables or views.

Most query languages allow unqualified column names if the column names are not ambiguous.

If a column name exists in more than one table in the query, you need to qualify the column name with a table name.

Qualifying Tables

To qualify a column in most SQL, use this syntax:

```
<database>.<ownerid>.<table/view>.<column>
```

PeopleSoft applications do not normally require the database or owner ID qualifications.

By the nature of a normalized relational database, most queries involve table joins.

For example, in the training database, the student ID is stored once in PSU_STUDENT_TBL, while related education data for the students is stored in many other tables.

SQL Outer Joins

To return all students, whether they are enrolled in a course or not, you must force an outer join.

This is the Query Analyzer syntax:

```
SELECT  A.STUDENT_NAME,
            B.COURSE,
FROM   PS_PSU_STUDENT_TBL A,
            PS_PSU_STU_ENROLL B
WHERE   A.STUDENT_ID = B.STUDENT_ID
ORDER BY B.COURSE
```

Using Subqueries in SQL Statements

In SQL, a subquery is a SELECT command that you enclose in parentheses in the WHERE clause. The database processes subqueries prior to the rest of the query to obtain values for the WHERE clause.

You sometimes call subqueries *subselects.*

The basic syntax is:

```
SELECT  <columns>
FROM  <table/view>
WHERE  <column> =
      (SELECT <matching column>
        FROM <table/view>
      WHERE <column> = <value>)
```

Use SQL Subqueries to enter selection criteria that is dynamic and based on field values.

For example, you retrieve the current employees who are also managers using this statement:

```
SELECT  NAME
FROM  EMPLOYEE_DATA
WHERE  EMPLID IN
         (SELECT DISTINCT MANAGERID
          FROM EMPLOYEE_DATA)
```

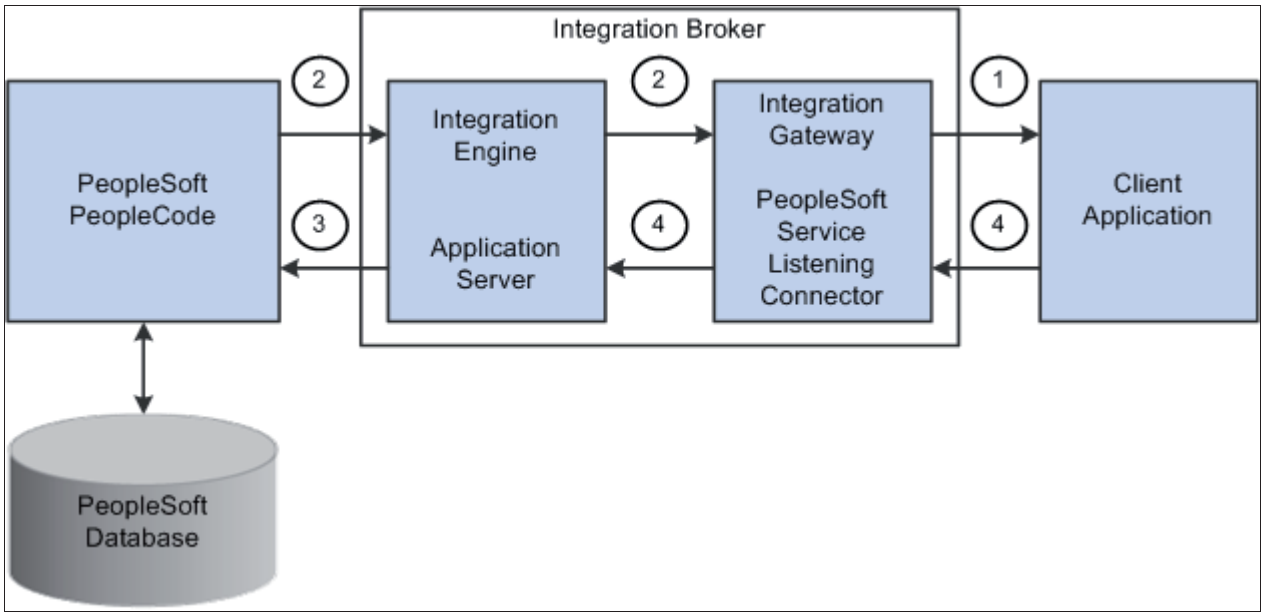
Note. The MANAGERID field contains the EMPLID value of the employee's manager.

Appendix D

Query Access Service (QAS)

Web Services Overview

Web Services support interoperable machine-to-machine interaction over a network. Web services in PeopleSoft are implemented through the PeopleTools Integration Broker (IB) framework. The Integration Gateway web application receives all the web service requests and forwards them to the Integration Engine (application server) for processing. This diagram illustrates how the web service is used with external applications.



This process occurs when you use a reporting web service with a client application:

1. The client application invokes one of the service operations from the API (application programming interface) published by PeopleSoft application.
2. PeopleSoft Integration Broker receives the service operation request and validates the WS security credentials.
3. The request is passed to Application Server for processing. The Application Server authenticates the service operation and routes it to the respective handler. The handler executes the PeopleCode and sends the response to Integration Gateway.
4. Integration Broker sends the response to the client application.

QAS Service Operations

To enable external applications to access PeopleSoft Query, a number of service operations are available. Depending on the application, the external application will need to use several service operations in combination. The service operations for QAS are part of the service QAS_QRY_SERVICE. QAS service operations are grouped into the following categories to help describe the service operations:

QAS service operations are grouped into the following categories to help describe the service operations:

- Query Creation Services (QCS)
- Query Security Services (QSS)
- Query Metadata Services (QMS)
- Query Execution Services (QES)

Note. No processing or security is dependent on the categories; they are simply groupings for describing the service operations.

Query Creation Services

Query creation services are used by client applications to create and save a new query. This table lists the Query Creation Services:

Service Operation Name	*Description*
QAS_RECORDS_OPER	Returns the list of records, along with descriptions, that are accessible to the user.
QAS_HIERARCHY_RECORDS_OPER	Returns the list of hierarchy records, if any, along with their descriptions.
QAS_RELATED_RECORDS_OPER	Returns the list of related records of all the fields in the record.
QAS_RECORD_DEFN_OPER	Returns the definition of a given record, including field descriptions and key information.
QAS_TREES_OPER	Returns the list of PeopleSoft trees that are accessible to the user.
QAS_TREE_DETAILS_OPER	Returns the tree details, including setID, effective date, and the list of all the nodes.
QAS_FIELDS_OPER	Returns the fields from all records that are accessible to the user.
QAS_FIELD_PROPS_OPER	Returns the properties of a given field.
QAS_QUERY_DETAILS_OPER	Returns the complete details of an existing query in XML format.
QAS_SAVE_QUERY_OPER	Validates and verifies XML-format query and saves the query in the PeopleSoft database.
QAS_QUERY_DELETE_OPER	Deletes the query from the PeopleSoft database.

Query Security Services

Query security services are used to identify users and roles. This table lists the Query Security Services:

Service Operation Name	Description
QAS_AUTHTOKEN_OPER	Returns the user ID based on the PS Token passed in.
QAS_LISTROLE_OPER	Returns the list of roles, along with descriptions.
QAS_LISTUSER_OPER	Returns the list of users, along with descriptions.
QAS_LISTROLEUSERS_OPER	Returns the list of users for a given role, along with descriptions.
QAS_LISTUSERROLES_OPER	Returns the list of roles for a given user, along with descriptions.
QAS_LOGIN_OPER	Enables the client application to sign on to the PeopleSoft database.

Query Metadata Services

Query metadata services are used to extract application data from the PeopleSoft database. This table lists the Query Metadata Services:

Service Operation Name	Description
QAS_LISTQUERY_OPER	Returns the list of queries, along with the query descriptions, and owner type.
QAS_LISTQUERYPROMPTS_OPER	Returns a list of prompts used in a specific query.
QAS_LISTQUERYFIELDS_OPER	Returns a list of fields for a given query.
QAS_GETXLAT_OPER	Returns the translate value for a given field.
QAS_GETPROMPTTABLEVALUES_OPER	Returns a list of field values for a given prompt.

Query Execution Services

Query Execution Services are used to run the query from a third-party application. This table lists service operations used to execute a query:

Service Operation Name	*Description*
QAS_CANCELQUERY_OPER	Returns the status of query cancellation.
QAS_QUERYSTATUS_OPER	Returns query execution status.
QAS_EXECUTEQRYSYNC_OPER	Returns query result.
QAS_EXECUTEQRYASYNC_OPER	Returns query result.
QAS_EXECUTEQRYSYNCPOLL_OPER	Runs the query through process scheduler and returns the query instance ID. The service operation QAS_GETQUERYRESULTS_OPER is used to retrieve the results.
QAS_GETQUERYRESULTS_OPER	Used to return the query results when the query is executed on the PeopleSoft database using QAS_EXECUTEQRYSYNCPOLL_OPER.

QAS Security

When using QAS, the same query security used online determines which fields and records a user can access. Additional security exists in the service operations.

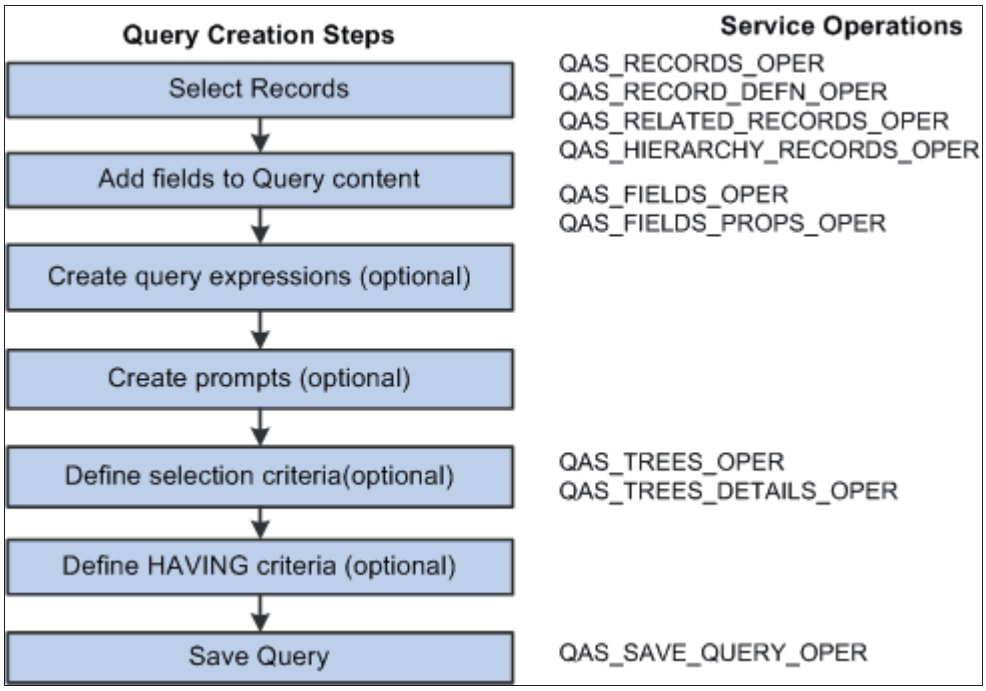
QAS service operations are delivered with User/Password Required enabled and WS Security Req Verification set to Encrypt and Digitally Sign or HTTPS. Client applications using QAS service operations must either digitally encrypt and sign the request or send the request over HTTPS.

Service operations are secured by means of permission lists. PeopleSoft applications deliver the permission list PTPT2200 (QAS access), which has full access to all QAS service operations and the application engine program QASEXEQRY. The role QAS Admin contains the permission list PTPT2200. Any users assigned the role QAS Admin can access the QAS service operations.

See *Enterprise PeopleTools 8.50 PeopleBook: Reporting Web Services* , "Accessing PeopleSoft Application Tables"

Creating a Query Using QAS

QAS provides several service operations that can be used by an external application to access the PeopleSoft application database to select records and fields to create a valid PeopleSoft Query. Creating a query involves several steps, some of which do not require access to the PeopleSoft records, fields, and trees and therefore do not have associated service operations. This diagram illustrates the steps and corresponding service operation that can be used to access the PeopleSoft database information:



You can use QAS_RECORDS_OPER, QAS_RECORD_DEFN_OPER, QAS_RELATED_RECORDS_OPER, and QAS_HIERARCHY_RECORDS_OPER to select records.

QAS_FIELDS_OPER and QAS_FIELD_PROPS_OPER can be used to help select the fields and prompts to add to the query content.

QAS_TREE_OPER and QAS_TREE_DETAILS_OPER can be used when your selection criteria uses trees.

The QAS_SAVE_QUERY_OPER is used to save a query definition to the PeopleSoft database.

Example Request to Save a Query

This is an example of the SQL for to list student IDs and names for a specific customer:

```
SELECT A.STUDENT_ID, A.STUDENT_NAME, A.CUSTOMER_ID
  FROM PS_PSU_STUDENT_TBL A
  WHERE A.CUSTOMER_ID = :1
  ORDER BY 1
```

This is the request to create and save this query as STUDENT_LIST:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:qas="http://xmlns.oracle.com/Enterprise/Tools/schemas/QAS_QUERY_SAV
E_REQ_MSG.VERSION_1">
   <soapenv:Header/>
   <soapenv:Body>
      <qas:QAS_QUERY_SAVE_REQ_MSG>
         <!--1 or more repetitions:-->
         <RECORD>
            <RCDNUM>1</RCDNUM>
            <RCDNAME>PSU_STUDENT_TBL</RCDNAME>
            <CORRNAME>A</CORRNAME>
            <JOINTYPE></JOINTYPE>
            <JOINRCDALIAS></JOINRCDALIAS>
            <JOINFIELD></JOINFIELD>
            <RCDSELECTNUM>1</RCDSELECTNUM>
         </RECORD>
         <!--1 or more repetitions:-->
         <FIELD>
            <FIELDNUM>1</FIELDNUM>
            <FIELDNAME>STUDENT_ID</FIELDNAME>
            <FIELDRCDNUM>1</FIELDRCDNUM>
            <DESCR></DESCR>
            <AGGREGATE_TYPE>None</AGGREGATE_TYPE>
            <HEADING_TYPE>RFT Short</HEADING_TYPE>
            <HEADING></HEADING>
            <COLUMNNUM>1</COLUMNNUM>
            <ORDERBYNUM>1</ORDERBYNUM>
            <ORDERBYDIR></ORDERBYDIR>
            <FIELDSELECTNUM>1</FIELDSELECTNUM>
            <EXPRESSION_AS_FIELD></EXPRESSION_AS_FIELD>
         </FIELD>
         <FIELD>
            <FIELDNUM>2</FIELDNUM>
            <FIELDNAME>STUDENT_NAME</FIELDNAME>
            <FIELDRCDNUM>1</FIELDRCDNUM>
            <DESCR></DESCR>
            <AGGREGATE_TYPE>None</AGGREGATE_TYPE>
            <HEADING_TYPE>RFT Short</HEADING_TYPE>
            <HEADING></HEADING>
            <COLUMNNUM>2</COLUMNNUM>
            <ORDERBYNUM>1</ORDERBYNUM>
            <ORDERBYDIR></ORDERBYDIR>
            <FIELDSELECTNUM>1</FIELDSELECTNUM>
            <EXPRESSION_AS_FIELD></EXPRESSION_AS_FIELD>
         </FIELD>
```

```
<FIELD>
   <FIELDNUM>3</FIELDNUM>
   <FIELDNAME>CUSTOMER_ID</FIELDNAME>
   <FIELDRCDNUM>1</FIELDRCDNUM>
   <DESCR></DESCR>
   <AGGREGATE_TYPE>None</AGGREGATE_TYPE>
   <HEADING_TYPE>RFT Short</HEADING_TYPE>
   <HEADING></HEADING>
   <COLUMNNUM>3</COLUMNNUM>
   <ORDERBYNUM>1</ORDERBYNUM>
   <ORDERBYDIR></ORDERBYDIR>
   <FIELDSELECTNUM>1</FIELDSELECTNUM>
   <EXPRESSION_AS_FIELD></EXPRESSION_AS_FIELD>
</FIELD>>
<!--Zero or more repetitions:-->
<CRITERION>
   <CRTNUM>1</CRTNUM>
   <CRTNAME>Cust</CRTNAME>
   <CRTHAVINGFLAG>False</CRTHAVINGFLAG>
   <CRTSELECTNUM>1</CRTSELECTNUM>
   <CRTNEGATION>False</CRTNEGATION>
   <CONDITION_TYPE>equal to</CONDITION_TYPE>
   <LEFT_PARENTHESIS_LEVEL>0</LEFT_PARENTHESIS_LEVEL>
   <RIGHT_PARENTHESIS_LEVEL>0</RIGHT_PARENTHESIS_LEVEL>
   <CRTEXP1TYPE>Field</CRTEXP1TYPE>
   <CRTEXP1TEXT></CRTEXP1TEXT>
   <CRTEXP1NUM></CRTEXP1NUM>
   <CRTEXP1RCDALIAS>A</CRTEXP1RCDALIAS>
   <CRTEXP1FIELD>CUSTOMER_ID</CRTEXP1FIELD>
   <CRTEXP2RCDALIAS>A</CRTEXP2RCDALIAS>
   <CRTEXP2FIELD></CRTEXP2FIELD>
   <CRTEXP2TYPE>prompt</CRTEXP2TYPE>
   <CRTEXP2TEXT>:1</CRTEXP2TEXT>
   <CRTLOGICALOPER>not used</CRTLOGICALOPER>
</CRITERION>
<!--Zero or more repetitions:-->
<PROMPT>
   <PROMPT_NUM>1</PROMPT_NUM>
   <PROMPT_NAME>CUSTOMER_ID</PROMPT_NAME>
   <PROMPT_UNIQUE_NAME>BIND1</PROMPT_UNIQUE_NAME>
   <PROMPT_FLDNAME>CUSTOMER_ID</PROMPT_FLDNAME>
   <PROMPT_TABLE>PSU_CUST_TBL</PROMPT_TABLE>
   <PROMPT_EDITTYPE>Prompt table</PROMPT_EDITTYPE>
   <PROMPT_HEADING>Customer</PROMPT_HEADING>
   <PROMPT_HEADINGTYPE>RFT Short</PROMPT_HEADINGTYPE>
   <PROMPT_FLDLENGTH>6</PROMPT_FLDLENGTH>
```

```
            <PROMPT_FLDDECIMALPOS>0</PROMPT_FLDDECIMALPOS>
         </PROMPT>
         <!--1 or more repetitions:-->
         <SELECT>
            <SELECTNUM>1</SELECTNUM>
            <PARENTSELECTNUM>0</PARENTSELECTNUM>
            <SELECTTYPE>Main</SELECTTYPE>
            <QRYDISTINCT>N</QRYDISTINCT>
         </SELECT>
         <QUERY_NAME>STUDENT_LIST</QUERY_NAME>
         <DESCRIPTION>Student List</DESCRIPTION>
         <DESCRLONG>This is an example of QAS</DESCRLONG>
         <QUERY_OWNER>Public</QUERY_OWNER>
      </qas:QAS_QUERY_SAVE_REQ_MSG>
   </soapenv:Body>
</soapenv:Envelope>
```

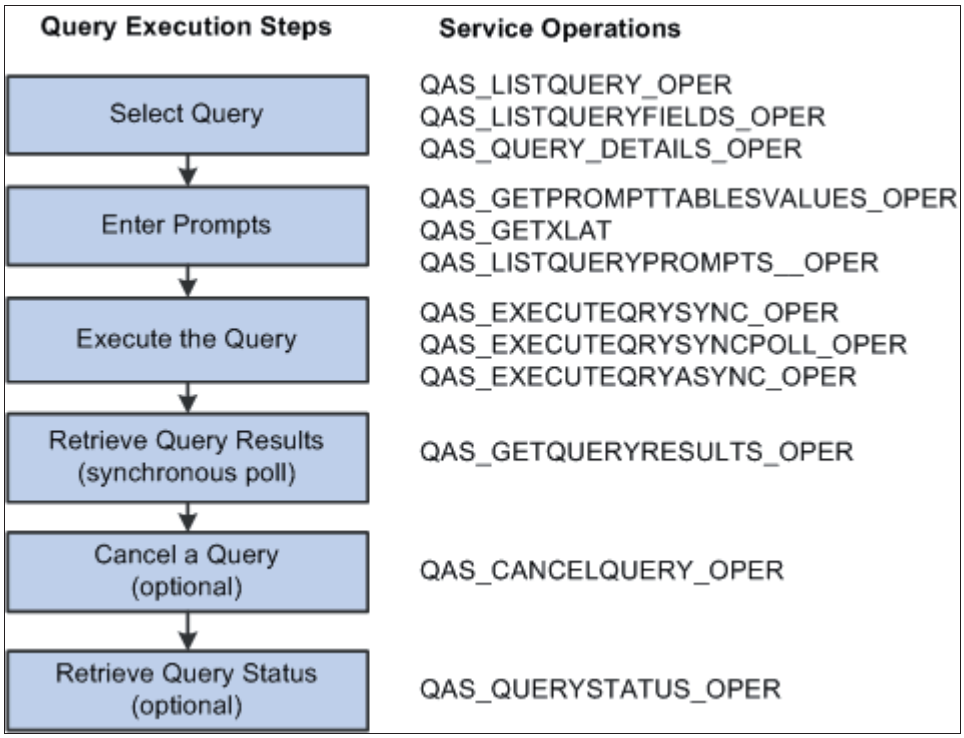
In order to create this query, the external application could use other QAS service operations to find the record names, field names, prompt values and so on. For example:

- Selecting the record. You can use the service operation QAS_RECORDS_OPER. to search for a record by RecordName, FieldName, or Description.
- Selecting fields from a record. You can use QAS_RECORD_DEFN_OPER to return the definition of a given record, including field descriptions and key information.

Note. For detailed information on using QAS, refer to *Enterprise PeopleTools 8.50 PeopleBook: Reporting Web Services*

Executing a Query Using QAS

To execute an existing query in the PeopleSoft database, the external application will use Query Metadata Services, Query Execution Services, and possibly the Query Result Service. This diagram illustrates the steps and corresponding service operations that can be used to execute a PeopleSoft query and retrieve the results:



To execute a query, you will need to know the query name. QAS_LISTQUERY_OPER, QAS_LISTQUERYFIELDS_OPER, and QAS_QUERY_DETAILS_OPER can be used to help select the query name.

For queries containing prompts, use QAS_GETPROMPTTABLEVALUES_OPER, QAS_GETXLAT_OPER, and QAS_LISTQUERYPROMPTS_OPER to identify the prompts and obtain a list of values.

You can also filter fields to be retuned in the query results using QAS_LISTQUERYFIELDS_OPER.

You can select how you want to execute the query and use the appropriate service operation. For synchronous poll execution, you will use QAS_GETQUERYRESULTS_OPER to retrieve the results.

You can invoke QAS_CANCELQUERY_OPER after obtaining the query result to clean up the PSQASRUN and IB tables.

You can check the query status using QAS_QUERYSTATUS_OPER.

Note. For detailed information on using QAS, refer to *Enterprise PeopleTools 8.50 PeopleBook: Reporting Web Services*
