COMMUNITY AND TECHNICAL COLLEGES
Washington State Board

QUERY LESSONS LEARNED REPORT

2022

Revisions

Change reference	Date
Original Version	12/05/2022

Contacts

For questions related to ctcLink reporting and the Lessons Learned Report, please email: dataservices@sbctc.edu

Table of Contents

Document Description

This document was created to provide information about findings and trends observed over the past year of query migrations; It contains examples of the most common pitfalls and recommendations to avoid them. Most queries submitted for migration are well developed and are migrated to production without any errors. Any queries that may be referenced in this document were chosen to highlight specific solutions and are not meant to single out any query, developer or institution.

FERPA Flag

If the query includes student name or any other identifiable information about a student, it must have the FERPA flag. The FERPA Flag is most often seen with name and should be placed immediately after the name.

Link to Data Brief - FERPA

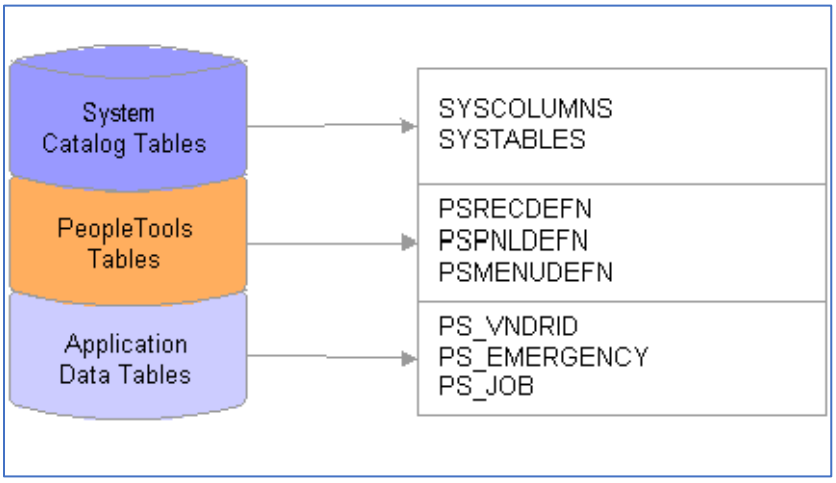
https://www.sbctc.edu/resources/documents/colleges-staff/commissions-councils/dgc/ferpa-directory-information.pdf

PeopleSoft Table Types

PeopleSoft applications are table-based systems that contains three major sets of tables:

- System Catalog tables store physical attributes of tables and views, which the database management system uses to optimize performance.
- PeopleTools tables contain information that you define using PeopleTools.
- Application Data tables house the actual data that users enter and access through PeopleSoft application pages.

This figure shows PeopleSoft database tables and sample names:



PeopleSoft Naming Conventions for Delivered Records

Suffixes used by PeopleSoft to identify certain types of record definitions

_TBL Identifies edit or prompt tables that contain data that is used for validation, as opposed to data that is maintained by the application.

- Prompt tables store commonly used values. They include, but are not limited to, control tables, which store institution-wide values. For example:
  - ACAD_PROG_TBL - Academic Program Table
  - ITEM_TYPE_TBL - Item Type Table
  - SRVC_IND_CD_TBL - Service Indicator Code Table

- ACAD_PLAN_TBL - Academic Plan Table
- TERM_VAL_TBL - Term Value Table

_VW Identifies record definitions that are physically implemented by defining an SQL view.

Example

- ACAD_PLAN_VW

_DVW Identifies a dynamic view. This is a record definition that can be used like a view in pages and PeopleCode but is not actually stored as an SQL view in the database. Instead, the system uses the view text as a base for the SQL select that is performed at runtime.

_WRK Identifies derived work records. These records are used for temporary storage and display. The data is not stored in the database.

_SBR Identifies subrecords.

- SELECT %subrec(JRNL_LN_KK_VW, A) FROM PS_JRNL_LN
- You may see this when using MetaLink to research the SQL script used to create a view

_QVW Identifies query views. a view that is constructed using the PeopleSoft Query tool. Before you can create the view, PeopleSoft Application Designer prompts you to save the definition.

_WL Identifies records as worklist record definitions.

Prefixes used by PeopleSoft to identify certain types of record definitions.

R_ Identifies work record definitions for Structured Query Report (SQR) reports. The remainder of the record name consists of the program or report ID.

AUDIT_ Identifies record definitions that store audit information for other record definitions in the database.

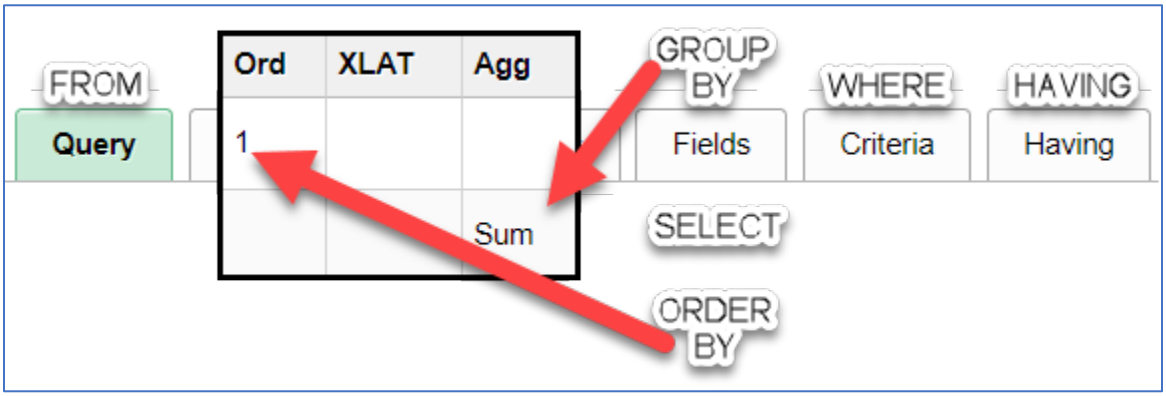
WEBLIB_ Identifies record definitions that store internet scripts.

FUNCLIB_ Identifies record definitions that contain written PeopleCode functions, as opposed to built-in functions.

DERIVED_ Identifies shared record definitions (across an application module or group) that have fields for PeopleCode events.

Query execution order and joins

When a query is submitted to the database, it is executed in the following order: FROM clause WHERE clause GROUP BY clause HAVING clause SELECT clause ORDER BY clause.



Why is it important to understand this?

- When a query is executed,
  - First all the tables and their join conditions from the where clause are executed filtering out invalid references between them.
  - Then the rest of the WHERE clause is applied which again filters the records based on the conditions given.
  - Now you have handful of records which are GROUP-ed
  - And HAVING clause is applied on the result.
  - As soon as it is completed, the columns mentioned are selected from the corresponding tables.
  - And finally sorted using ORDER BY clause.

So, when a query is written it should be verified based on this order, otherwise it will lead wrong result sets.

The join order within the FROM clause does matter because if we can join two tables that will reduce the number of rows needed to be processed by subsequent steps then our performance will improve.

As a best practice you should try to order your table join so the records joined first are the ones that reduce the result set the most.

1. Smaller is better
2. Your first record should provide the biggest bang for the buck.
   a. What record contains most of the data needed for the WHERE clause and the SELECT clause?
3. Understand your data and the table hierarchy
   a. Parent table
   b. Child table
4. Never start with the VCS_BIO views.
   a. This is a very large reference table and should not be used as a primary table

How to find a table/field name from a page

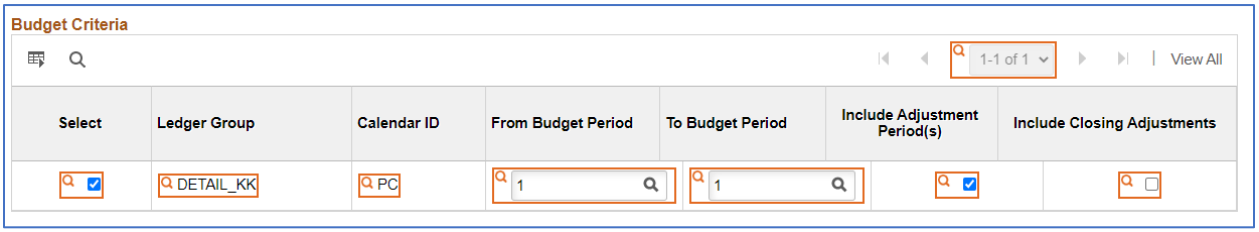
You will first need to have access to PS Utilities. PS Utilities is a Google Chrome Extension for use with PeopleSoft systems. It's useful for all users of Oracle PeopleSoft applications, whether end user, functional super user, or developer.

- It has many features, each of which can be turned on/off on the features page of PS Utilities Options.
- The PS Utilities bar will show up in the upper left of your browser when you go to a PeopleSoft page. This bar will have a help icon which you can use to find out more details about each of the features.

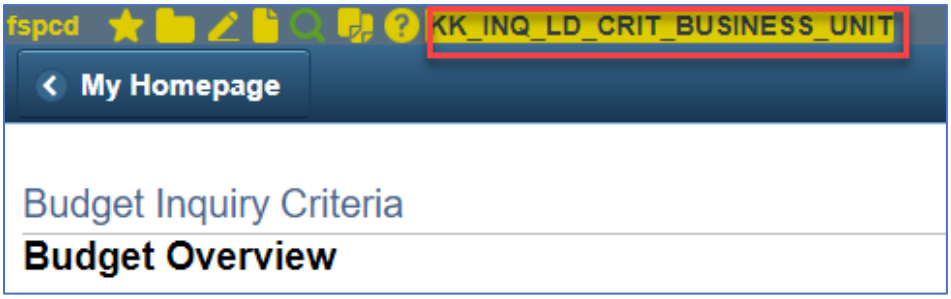
https://chrome.google.com/webstore/detail/ps-utilities/jajoopnifcliapcngocgiidifkmboemc?hl=en

Open a PeopleSoft page. Click on the magnifying glass icon from the PS Utilities bar. This is the Field Inspector.

Each field on the page will be surrounded by a red box with a magnifying glass in the left corner of each box.

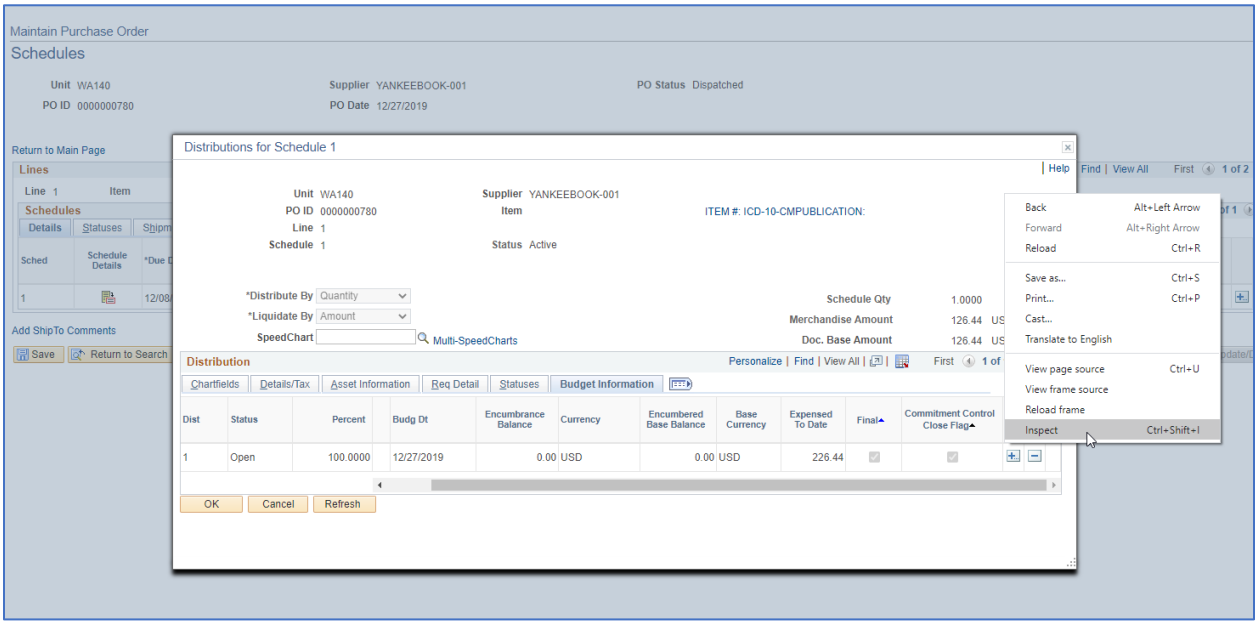


Hover the mouse over the magnifying glass next to each field and you will see the name of the table and field. If you want to "lock" the field name, click the magnifying glass which will turn it green, and the name of the table and field will stay at the top of the page.

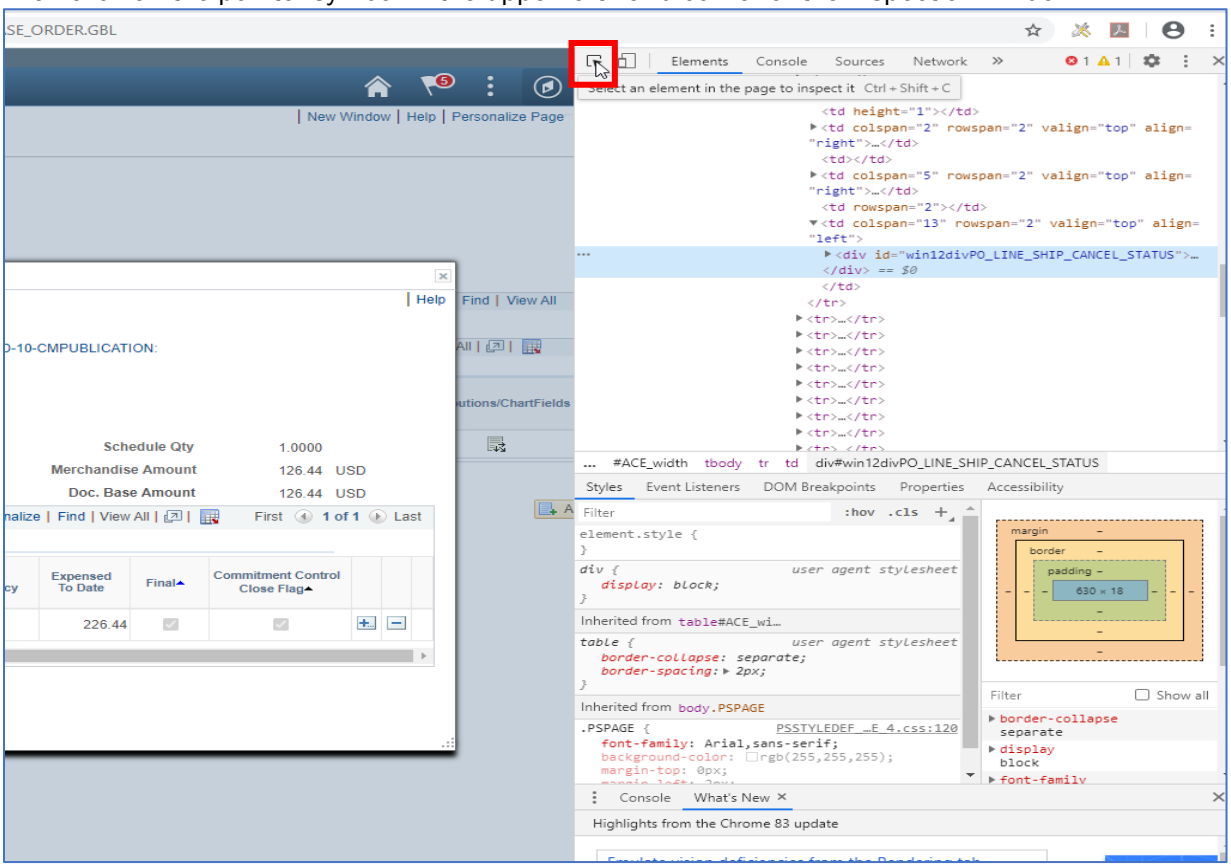


How to find a table/field name from a popup page

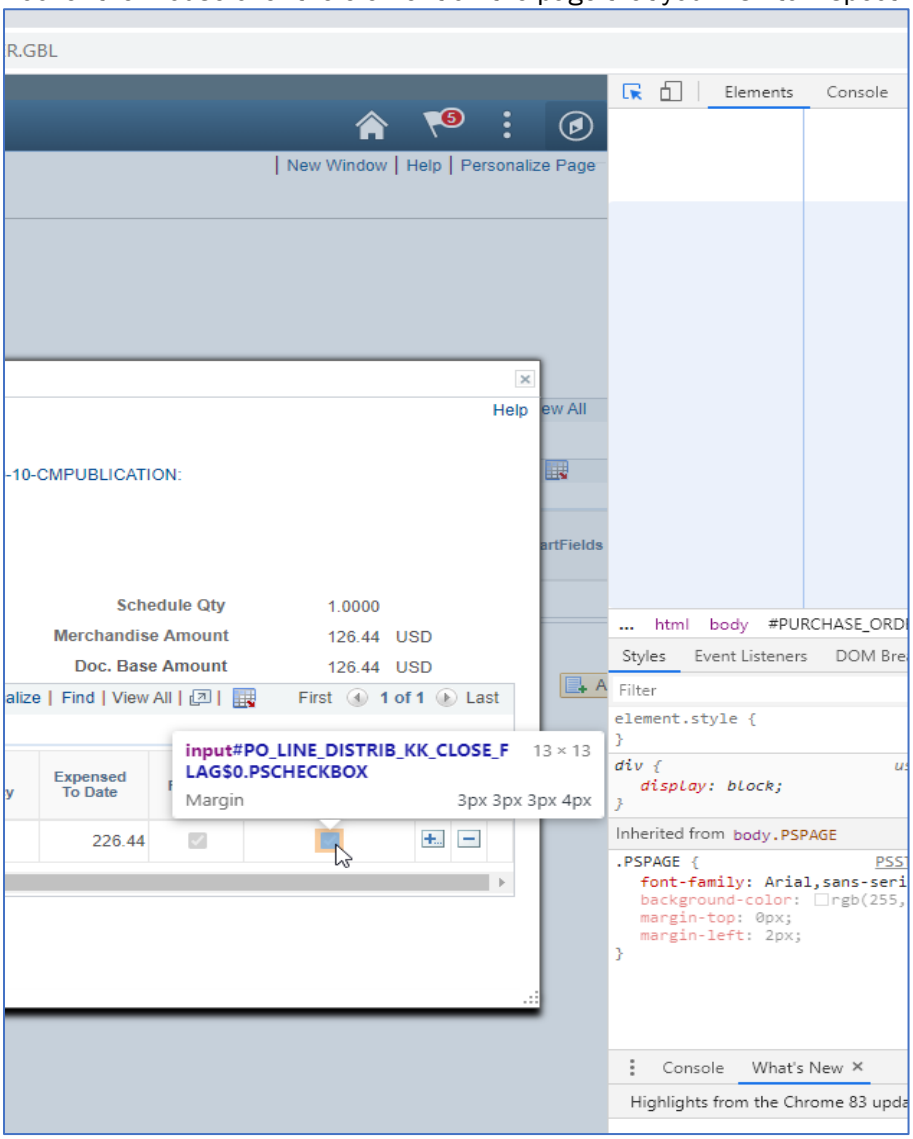
Hold your mouse/pointer over the pop-up page and right click. Click on “Inspect”



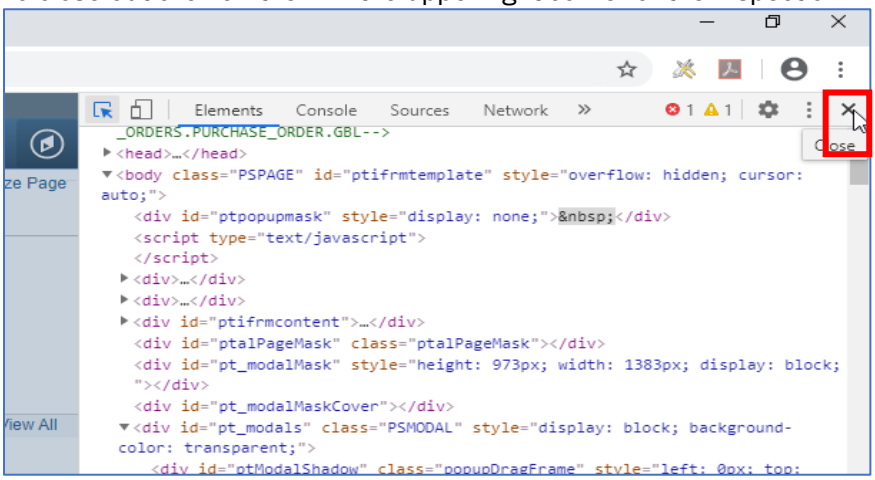
Then click on the pointer symbol in the upper left-hand corner of the inspection window.



Hoover the mouse over the element on the page that you wish to inspect.

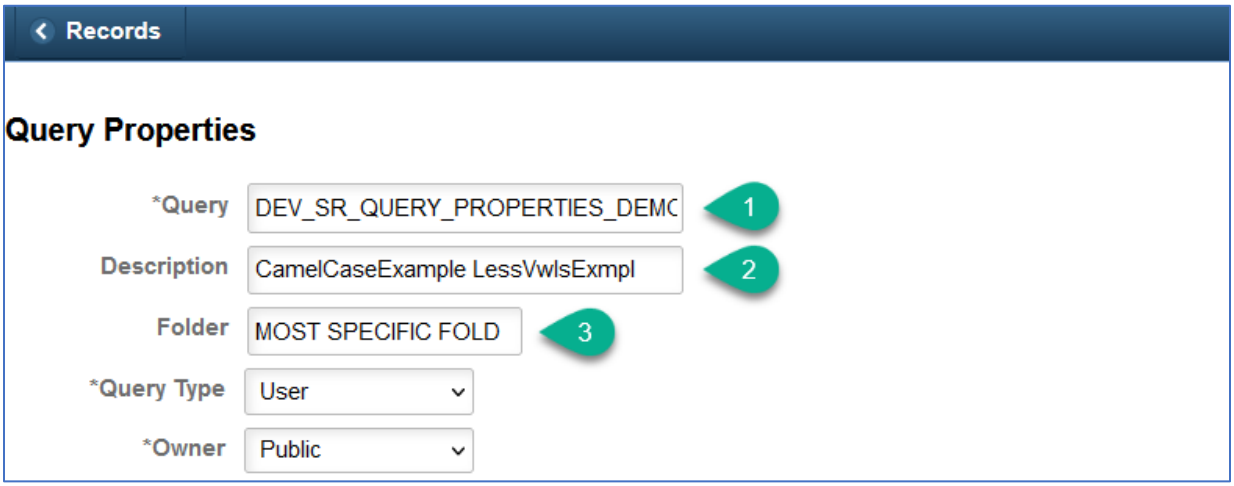


To close out click on the X in the upper right corner of the inspection window.



Query Properties

Entering information into the query properties page is an important step in development that can easily be overlooked or rushed. There were many lessons learned that revolve around the Query Properties Page.



Saving with the DEV_ prefix (1)

Saving your query with the DEV_ prefix provides two main benefits. The first benefit is the query will not be lost in the PCD refresh that happens every month. The second benefit is that it allows for a more thorough comparison between a modified version and the original.

Query Description (2)

Entering new description

When modifying or saving a query as a new query, it is essential to change the description. The query description is the first place many end users go to see information about the query, and if the information is the same across multiple queries, it can cause confusion.

Using CamelCase or Less Vowels

With the limited number of characters available to use in the description, there are methods that you can employ to maximize the space available while still being able to functionally read it. The use of abbreviations and acronyms is encouraged as well but be sure to include what they stand for in the definition. The first method in maximizing space is to remove the spaces in-between words, when this is done capitalize each word to help differentiate between them. This is called CamelCase. The next method is to remove vowels that are easily implied.

Query Folder (3)

Sometimes it's difficult to determine the best folder to add a query to. When you find yourself looking at the Pillar-Folder List https://www.sbctc.edu/resources/documents/colleges-staff/data-services/peoplesoft-ctclink/pillar-folder-lists.pdf and if there are multiple folders that seem appropriate, try to choose the most specific option you can. For example if your query is a population select that searches student records using a student group prompt, the query technically would work in the STUDENT RECORDS, STUDENT GROUPS, and the POPULATION SELECT folder. The most specific folder of these three would be the POPULATION SELECT folder.

Some folders should aways be selected:

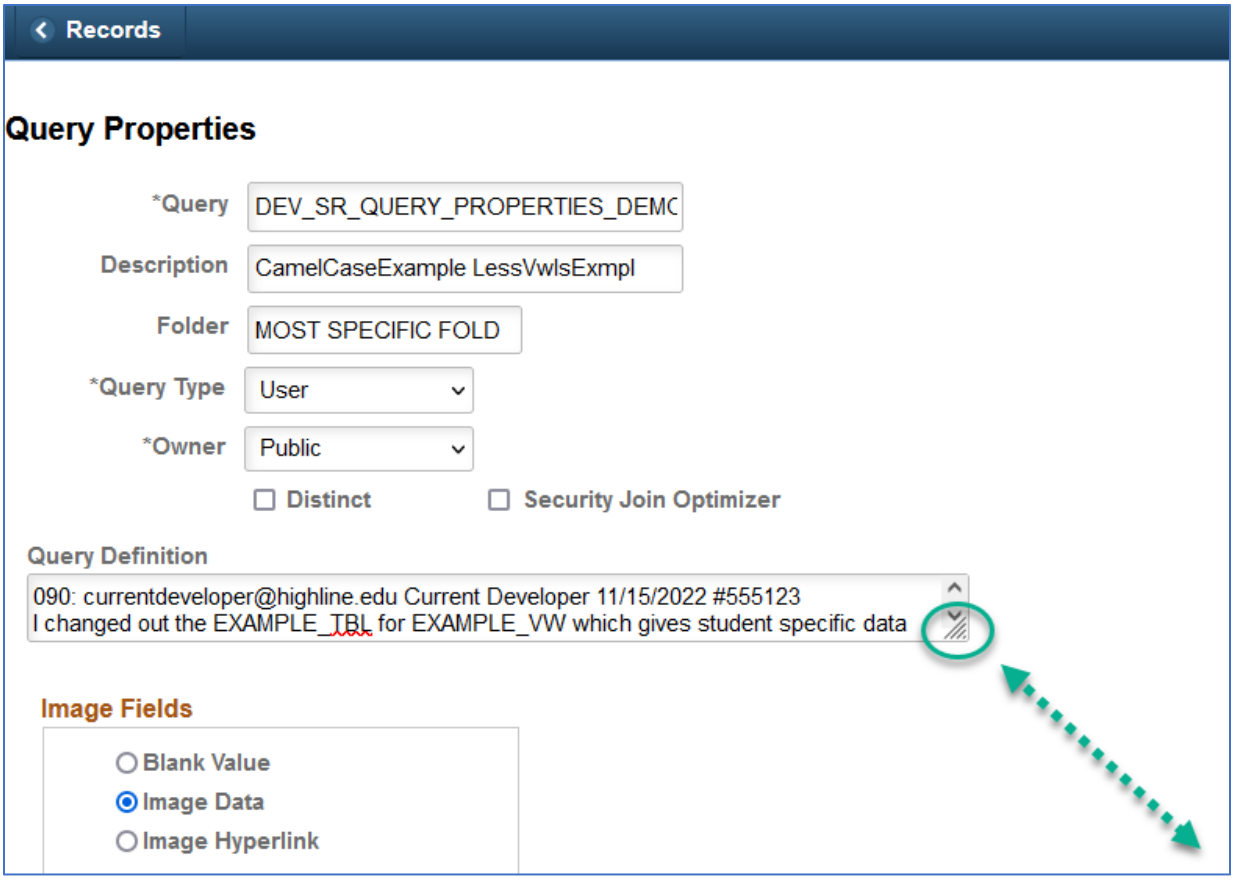
POPULATION SELECT for any population select queries

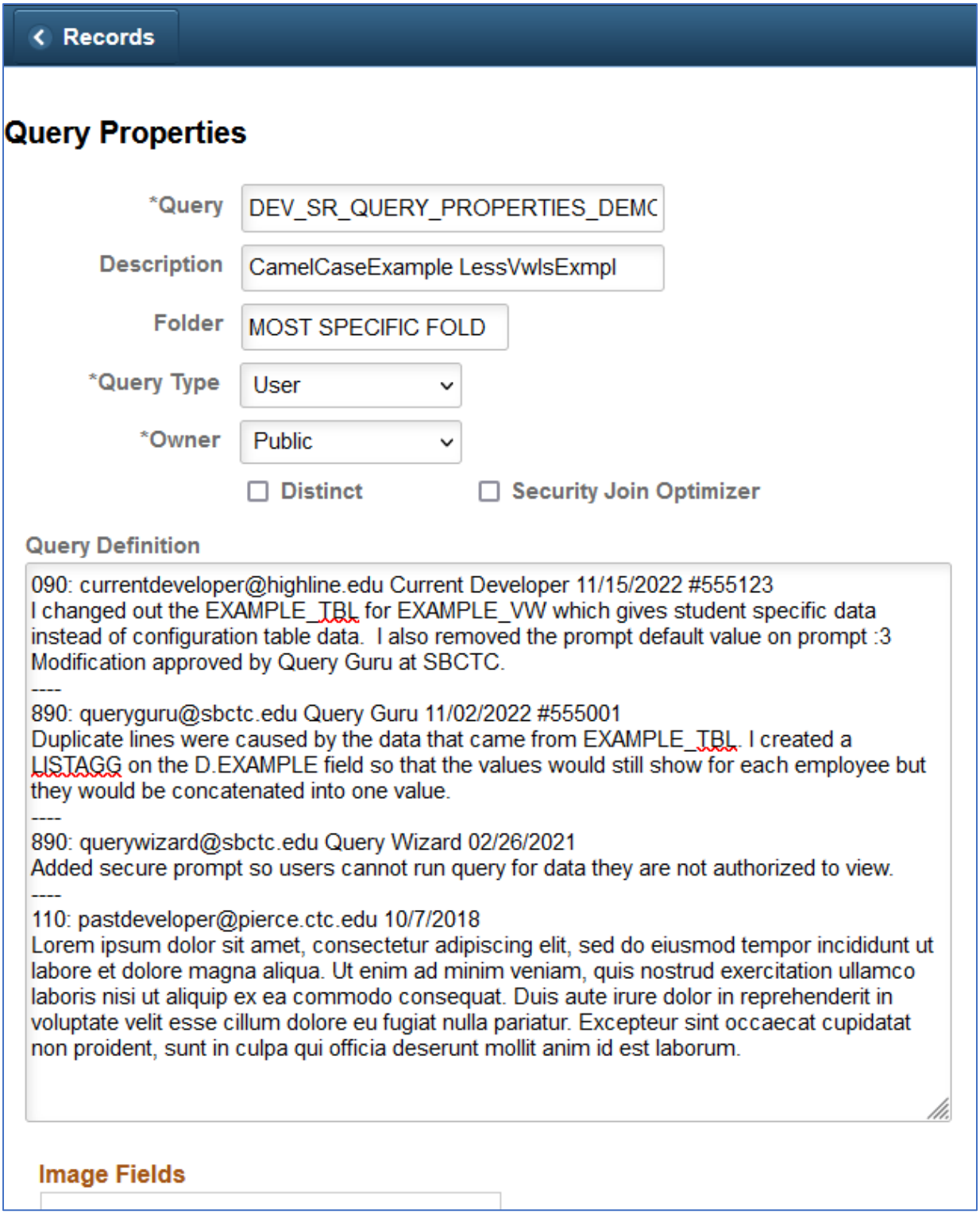
FINANCIAL AID for any Financial Aid queries

Query Definition

Expanding the Definition section

Clicking the bottom right corner and dragging down and to the right will expand the size of the definition box and allows for a much easier reading experience.

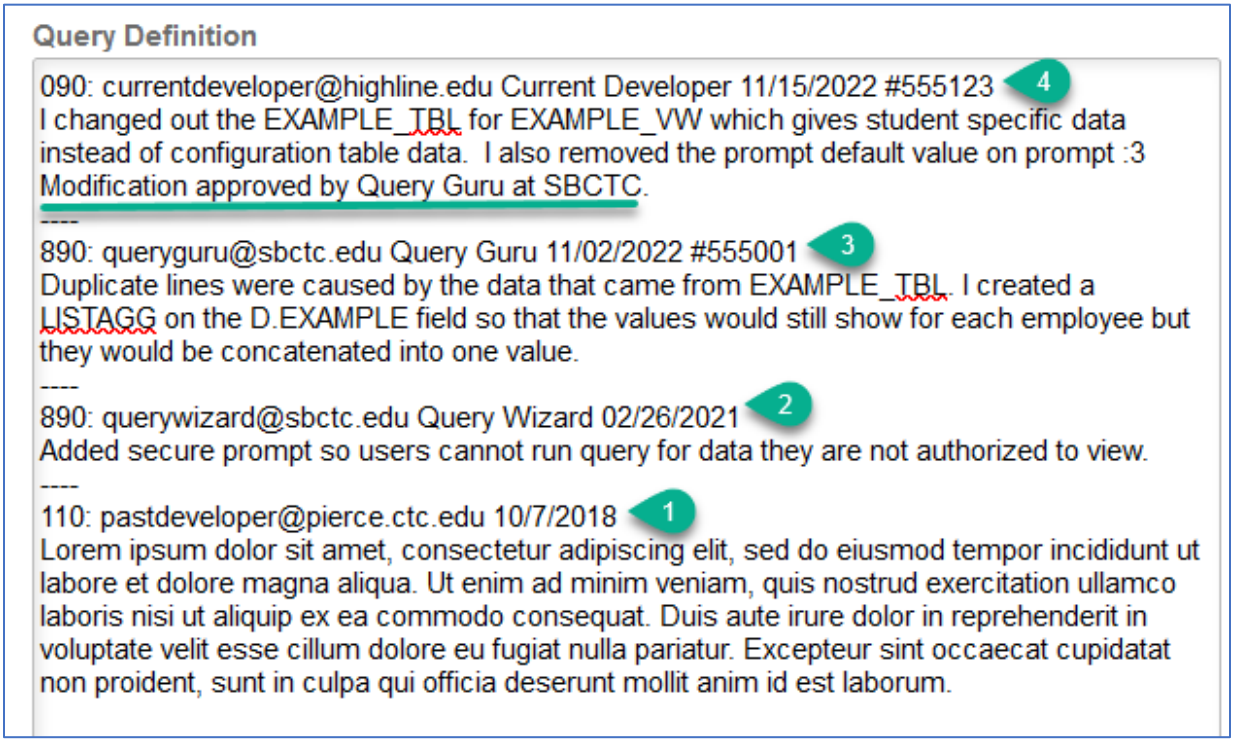

Records
Query Properties
*Query
DEV_SR_QUERY_PROPERTIES_DEMC
Description
CamelCaseExample LessVwlsExmpl
Folder
MOST SPECIFIC FOLD
*Query Type
User
*Owner
Public
Distinct
Security Join Optimizer
Query Definition
090: currentdeveloper@highline.edu Current Developer 11/15/2022 #555123
I changed out the EXAMPLE_TBL for EXAMPLE_VW which gives student specific data
Image Fields
Blank Value
Image Data
Image Hyperlink


Records
Query Properties
*Query
DEV_SR_QUERY_PROPERTIES_DEMC
Description
CamelCaseExample LessVwIsExmpl
Folder
MOST SPECIFIC FOLD
*Query Type
User
*Owner
Public
Distinct
Security Join Optimizer
Query Definition
090: currentdeveloper@highline.edu Current Developer 11/15/2022 #555123
I changed out the EXAMPLE_TBL for EXAMPLE_VW which gives student specific data instead of configuration table data. I also removed the prompt default value on prompt :3
Modification approved by Query Guru at SBCTC.

890: queryguru@sbctc.edu Query Guru 11/02/2022 #555001
Duplicate lines were caused by the data that came from EXAMPLE_TBL. I created a LISTAGG on the D.EXAMPLE field so that the values would still show for each employee but they would be concatenated into one value.

890: querywizard@sbctc.edu Query Wizard 02/26/2021
Added secure prompt so users cannot run query for data they are not authorized to view.

110: pastdeveloper@pierce.ctc.edu 10/7/2018
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Image Fields



Locating the first developer (1)

The very bottom entry in the query definition is the first person to develop the query. There are many older queries that may not have a lot of information in the first entry. Ideally there will be an email address and/or a name along with the first definition entry.

Definition can be found in Query Manager and metaLink.

Importance of identifying first developer

Whenever you are trying to modify a query, contacting the first developer is essential to ensure that you are not changing the expected result set that is already a part of the original developer's business processes.

If the original developer is not reachable, then please reach out to the SBCTC reporting team, preferably a team member who already has worked on the query (2 & 3)

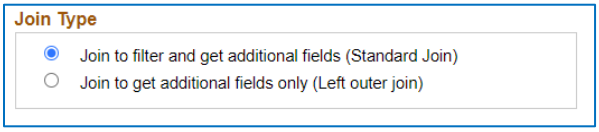
Adding who approved modification to definition update (4)

When you obtain approval to modify a query, please enter in the definition who authorized the modification.

Join Type Errors

Most queries utilize multiple records joined together. Knowing when to use which join and how to structure the criteria for them are important aspects of query development. Many issues we find have to do with using the wrong join type or not building the criteria correctly for the join.

The Two Join Types



Fruit	Citrus
Banana	No
Apple	No
Orange	Yes
Lemon	Yes

Fruit	Color
Banana	Yellow
Apple	Red
Apple	Green
Apple	Yellow

Standard Join (Inner Join)

A standard join only shows results where a match occurs between the two joined sets of data. This means that the query will display less results if the joined record does not have a match for every line of data in the record it is being joined to.

Fruit	Citrus	Color
Banana	No	Yellow
Apple	No	Red
Apple	No	Green
Apple	No	Yellow

The red Color table inner/standard joined to the blue Citrus table.

Left Outer Join

A left outer join adds additional data from the joined record without filtering the query where there is not a match to the record it is being joined to. Instead of restricting the results where there is not a match, the joined record contains no results (is null/blank) for the fields they add to the query.

Fruit	Citrus	Color
Banana	No	Yellow
Apple	No	Red
Apple	No	Green
Apple	No	Yellow
Orange	Yes	
Lemon	Yes	

The red Color table left outer joined to the blue Citrus table.

Note: Left outer joins use more system resources than inner joins. Therefore, if a query will get the

same results with either join, the inner/standard join is the one that should be used.

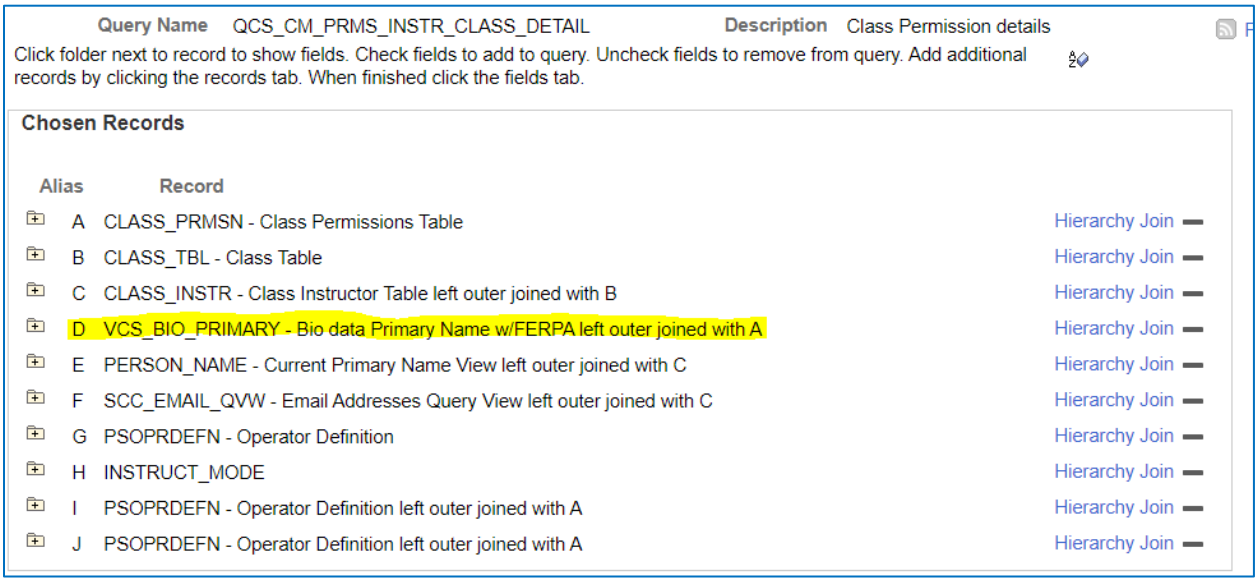
VCS_BIO_ Records

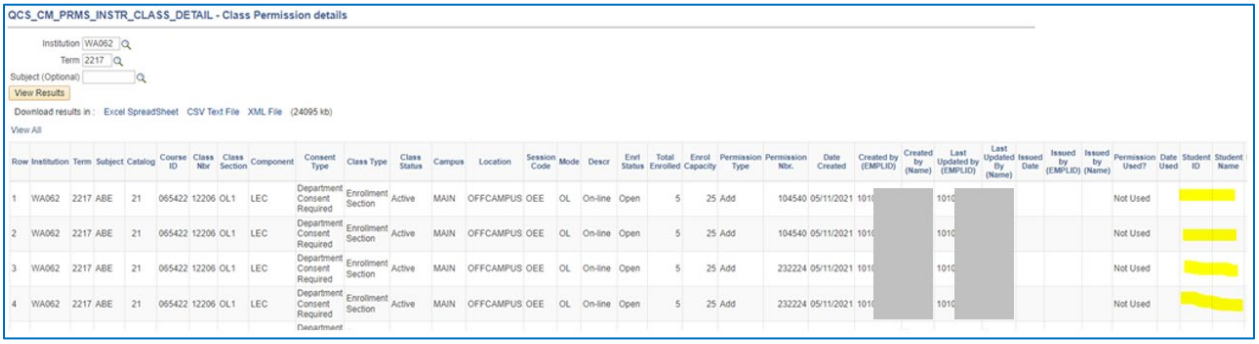
The VCS_BIO_PRIMARY and VCS_BIO_PRIORTY records are commonly used CS Pillar views that provide unduplicated listings of core student bio/demo data by EMPLID. Every student with an EMPLID will have a row of data in these views. Therefore, these should always be standard/inner joined to a query, not left outer joined. These are robust views that can be very taxing on performance when left outer joined.

Having one of these views left outer joined to a query is a common error that we run into. In many cases rebuilding the query as an inner join significantly increases performance.

If you are using a left outer join to tie one of these views to your query, make sure you understand and clearly explain in your migration request why this needs to be a left outer vs an inner join.

For example: In the query QCS_CM_PRMS_INSTR_CLASS_DETAIL the VCS_BIO_PRIMARY record is left outer joined because the EMPLID field in the CLASS_PRMSN record it is joined to is only populated after/if the permission code is used. The developer wanted the query to also show results for rows where the permission code is not being used. Therefore a left outer join was the best option:





Common Join Errors: Query Building Exercise

For this section I built a simple class information query to show some of the common join errors we encounter.
I created the query based on the CLASS_TBL (Class Table) record and used the first 12 fields of this record with a prompt for Institution and Term. For these examples I used WA062 as the institution value and 2233 for the term. Running this query for these values in PCP I got 368 results:

Institution = WA062, Term=2233

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

First 1-100 of 368 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Subject	Catalog	Career	Descr	Class Nbr
1	070491	1	2233	1	1L	WA062	SKILL	WTC	134	UGRD	Marine Mechanical	5044
2	070514	1	2233	1	1	WA062	SKILL	WTC	172	UGRD	Forklift Operation & Crt	5107
3	070490	1	2233	1	1	WA062	SKILL	WTC	133	UGRD	Wooden Boat Repair	5041

Error 01: Assuming There is a Data Match

Next, I wanted to add the following fields from the CLASS_INSTR record to the query:

1) CLASS_MTG_NBR (Class Meeting Pattern Nbr)
2) INSTR_ASSIGN_SEQ (Class/Instructor Assign Seq#)
3) EMPLID (ID of instructor)
4) INSTR_ROLE (Instructor Role)
5) ASSIGN_TYPE (Assignment Type)

Since every class has an instructor, I assumed there will be matching data in the CLASS_INSTR record for every row in the CLASS_TBL record and so used an inner join to bridge these two tables. However when I ran the query, I only received 218 results:

Institution = WA062, Term=2233

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

First 1-100 of 218 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Subject	Catalog	Career	Descr	Class Nbr	Pat Nbr	Assign Seq #	ID	Role	Assign Type
1	070491	1	2233	1	1L	WA062	SKILL	WTC	134	UGRD	Marine Mechanical	5044	1	1	201	PI	FT6
2	070514	1	2233	1	1	WA062	SKILL	WTC	172	UGRD	Forklift Operation & Crt	5107	1	1	101	PI	AD5
3	070491	1	2233	1	1	WA062	SKILL	WTC	134	UGRD	Marine Mechanical	5043	1	1	201	PI	FT6

Redoing this as a left outer join I get 380 results - slightly more than the 368 results I received before the join because some of the classes are taught by more than one instructor (adding additional rows):

Institution = WA062, Term=2233

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

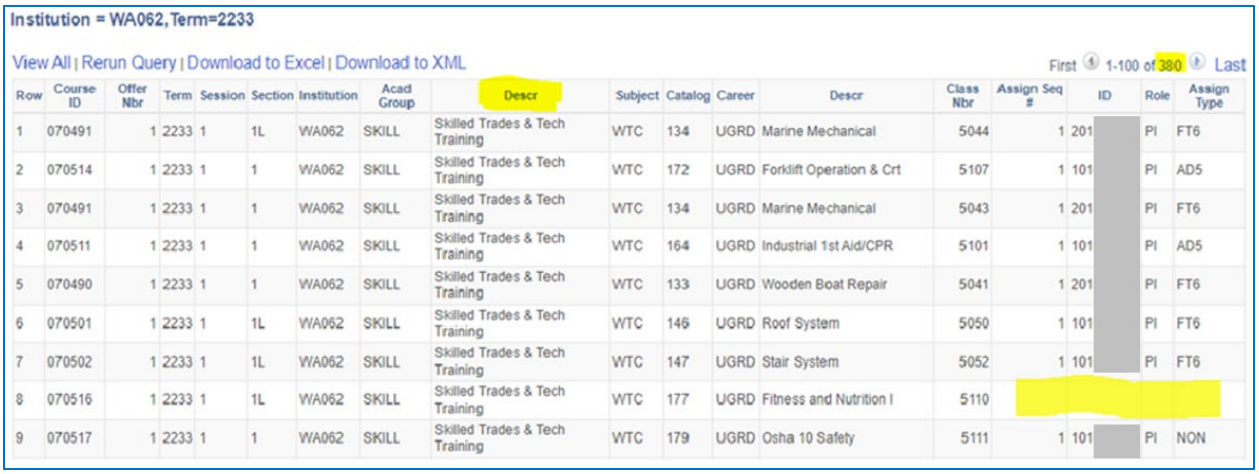
First 1-100 of 380 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Subject	Catalog	Career	Descr	Class Nbr	Pat Nbr	Assign Seq #	ID	Role	Assign Type
1	070491	1	2233	1	1L	WA062	SKILL	WTC	134	UGRD	Marine Mechanical	5044	1	1	201	PI	FT6
2	070514	1	2233	1	1	WA062	SKILL	WTC	172	UGRD	Forklift Operation & Crt	5107	1	1	101	PI	AD5
3	070491	1	2233	1	1	WA062	SKILL	WTC	134	UGRD	Marine Mechanical	5043	1	1	201	PI	FT6
4	070511	1	2233	1	1	WA062	SKILL	WTC	164	UGRD	Industrial 1st Aid/CPR	5101	1	1	101	PI	AD5
5	070490	1	2233	1	1	WA062	SKILL	WTC	133	UGRD	Wooden Boat Repair	5041	1	1	201	PI	FT6
6	070501	1	2233	1	1L	WA062	SKILL	WTC	146	UGRD	Roof System	5050	1	1	101	PI	FT6
7	070502	1	2233	1	1L	WA062	SKILL	WTC	147	UGRD	Stair System	5052	1	1	101	PI	FT6
8	070516	1	2233	1	1L	WA062	SKILL	WTC	177	UGRD	Fitness and Nutrition I	5110					

You can see from the blank cells in row eight of the example above that not all classes for this institution/term combination have an assigned instructor. Doing an inner join because you assume there will be a match between records is another common error we run into. It is important to know your data and not make assumptions. Errors like these can be avoided by exploring the data before joining it to a query, and then also checking to see if you get the expected results after the join.

Error 02: Unnecessary Left Outer Join

This is the opposite of the error demonstrated above. I wanted to add the academic group description field next to the Academic Group column and so I added the ACAD_GROUP_TBL record using a left outer join. I got the expected number of results and the query worked as intended. However, I noticed that there were no blank values in the new Description field like there had been for the CLASS_INSTR fields in the previous example:



Institution = WA062,Term=2233

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

First 1-100 of 380 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Descr	Subject	Catalog	Career	Descr	Class Nbr	Assign Seq #	ID	Role	Assign Type
1	070491	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5044	1	201	PI	FT6
2	070514	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	172	UGRD	Forklift Operation & Crt	5107	1	101	PI	AD5
3	070491	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5043	1	201	PI	FT6
4	070511	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	164	UGRD	Industrial 1st Aid/CPR	5101	1	101	PI	AD5
5	070490	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	133	UGRD	Wooden Boat Repair	5041	1	201	PI	FT6
6	070501	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	146	UGRD	Roof System	5050	1	101	PI	FT6
7	070502	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	147	UGRD	Stair System	5052	1	101	PI	FT6
8	070516	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	177	UGRD	Fitness and Nutrition I	5110				
9	070517	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	179	UGRD	Osha 10 Safety	5111	1	101	PI	NON

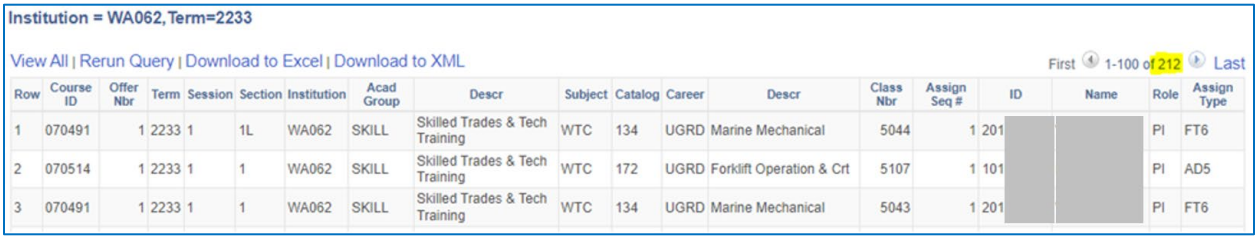
Looking further into these records, I discovered that every row in the CLASS_TBL record has an Academic Group (ACAD_GROUP) value and each of these has a corresponding value in the ACAD_GROUP_TBL.

In this example the join therefore should be an inner join. Remember - When either join produces the wanted results a left outer join should not be used. This is a common mistake because the query produces the same results either way, though doing a left outer join instead of an inner join wastes resources and can possibly cause issues or confusion if the query needs modified later.

Note: Records that end in _TBL are edit or prompt tables that contain much of the fundamental data often used throughout ctcLink as the foundation for more complex records or views. Therefore these will usually be inner joined in a query as opposed to left outer joined. Many of the recommended prompt tables are _TBL records.

Error 03: An Inner Join on a Left Outer Join

Next, I wanted to add Instructor Name next to the Instructor ID field. To get this data I joined the PERSON_NAME record to the CLASS_INSTR record. Because there should be a record for every EMPLID in the PERSON_NAME record I structured this as an inner join. However, when I ran the query I only got 212 results instead of the expected 380:



Institution = WA062, Term=2233

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

First 1-100 of 212 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Descr	Subject	Catalog	Career	Descr	Class Nbr	Assign Seq #	ID	Name	Role	Assign Type
1	070491	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5044	1	201		PI	FT6
2	070514	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	172	UGRD	Forklift Operation & Crt	5107	1	101		PI	AD5
3	070491	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5043	1	201		PI	FT6

This is because doing an inner join puts the criteria on the where clause which means the entire result set is restricted to where a match exists between all the involved data. Redoing this as a left outer join I get the expected number of results:

Institution = WA062, Term=2233

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

First 1-100 of 380 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Descr	Subject	Catalog	Career	Descr	Class Nbr	Assign Seq #	ID	Name	Role	Assign Type
1	070491	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5044	1	201		PI	FT6
2	070514	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	172	UGRD	Forklift Operation & Crt	5107	1	101		PI	AD5
3	070491	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5043	1	201		PI	FT6
4	070511	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	164	UGRD	Industrial 1st Aid/CPR	5101	1	101		PI	AD5
5	070490	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	133	UGRD	Wooden Boat Repair	5041	1	201		PI	FT6
6	070501	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	146	UGRD	Roof System	5050	1	101		PI	FT6
7	070502	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	147	UGRD	Stair System	5052	1	101		PI	FT6
8	070516	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	177	UGRD	Fitness and Nutrition I	5110					

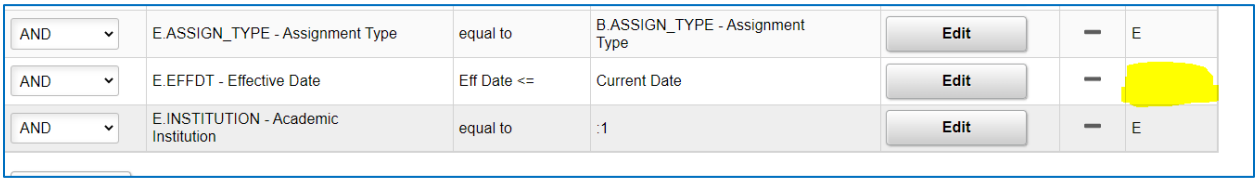
A record should never be inner joined to a left outer joined record. Doing so functionally turns that previous left outer join into an inner join. An inner join or a left outer join can both be added onto another inner joined record, however only a left outer join should be added onto another left outer join.

Error 04: Left Outer Joined EFFDT criteria Placed on the Where Clause

Next I wanted to add a description of the instructor assignment type (ASSIGN_TYPE) to the query. To do this I joined the ASSIGNMENT_TYPE table to the CLASS_INSTR table. Since the CLASS_INSTR record is joined to the query using a left outer join, I know that the ASSIGNMENT_TYPE table should also be joined using a left outer join.

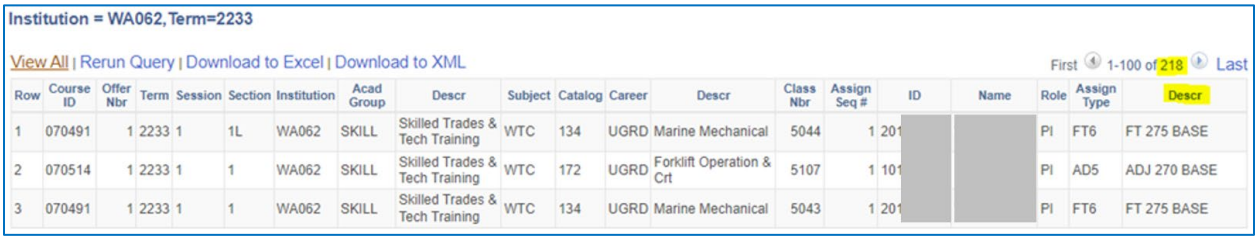
Note: Since INSTITUTION is a key field in the ASSIGNMENT_TYPE record but there is no INSTITUTION in the CLASS_INSTR table I added criteria making the CLASS_INSTR INSTITUTION equal to :1 (the Intsitution Prompt). Since I added this record as a left outer join this criteria needs to be put on the outer join clause, not the where clause.

When I finished adding this criteria, I noticed that the Effective Date criteria was on the where clause:



AND	E.ASSIGN_TYPE - Assignment Type	equal to	B.ASSIGN_TYPE - Assignment Type	Edit	−	E
AND	E.EFFDT - Effective Date	Eff Date <=	Current Date	Edit	−	
AND	E.INSTITUTION - Academic Institution	equal to	:1	Edit	−	E

If I leave this as is and run the query, I will only get 218 results:

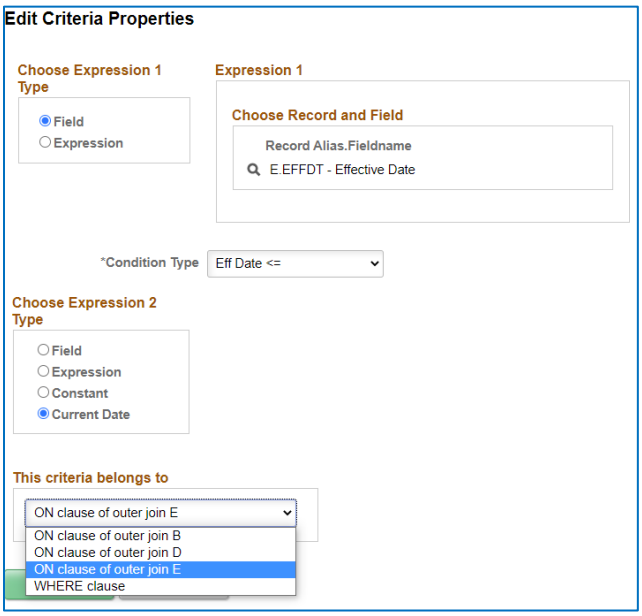


Institution = WA062, Term=2233

View All | Rerun Query | Download to Excel | Download to XML — First 1-100 of 218 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Descr	Subject	Catalog	Career	Descr	Class Nbr	Assign Seq #	ID	Name	Role	Assign Type	Descr
1	070491	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5044	1	201		PI	FT6	FT 275 BASE
2	070514	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	172	UGRD	Forklift Operation & Crt	5107	1	101		PI	AD5	ADJ 270 BASE
3	070491	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5043	1	201		PI	FT6	FT 275 BASE

To fix this I will need to edit the Effective Date criteria so that it belongs to the left outer join clause:



After doing so when I ran the query again, I received the expected 380 results:

Institution = WA062, Term=2233

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

First 1-100 of 380 Last

Row	Course ID	Offer Nbr	Term	Session	Section	Institution	Acad Group	Descr	Subject	Catalog	Career	Descr	Class Nbr	Assign Seq #	ID	Name	Role	Assign Type	Descr
1	070491	1	2233	1	1L	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5044	1	201		PI	FT6	FT 275 BASE
2	070514	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	172	UGRD	Forklift Operation & Crt	5107	1	101		PI	AD5	ADJ 270 BASE
3	070491	1	2233	1	1	WA062	SKILL	Skilled Trades & Tech Training	WTC	134	UGRD	Marine Mechanical	5043	1	201		PI	FT6	FT 275 BASE

Though the ASSIGNMENT_TYPE record was left outer joined, having the EFFDT criteria on the where clause made the join behave as an inner join because every row where the EFFDT was blank/null was excluded from the results instead of simply displaying blank/null values for the Descr field.

This is a very easy mistake to make because regardless of the type of join you are doing, Peoplesoft always puts the automated Effective Date criteria on the where clause.

Note: The ability to add effective date criteria to the outer join clause is a fairly new function in ctcLink. Before this was available as a feature the old way of coding effective date criteria for a left outer join was to add an "or null" clause to the criteria and group it with the other EFFDT where clause criteria:

AND	E.INSTITUTION - Academic Institution	equal to	:1	Edit	−	E
AND	(E.EFFDT - Effective Date	Eff Date <=	Current Date	Edit	−	
OR	E.EFFDT - Effective Date	is null	)	Edit	−	

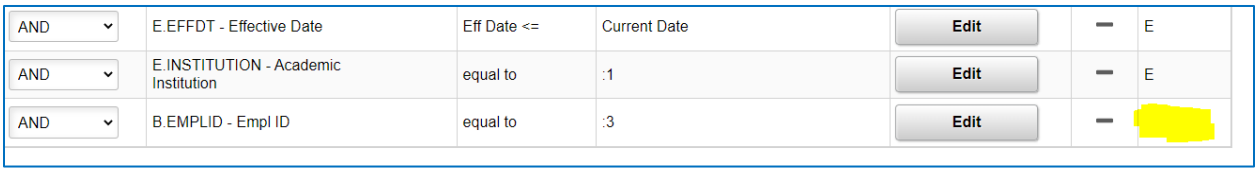
Though newly created or modified queries should use the new EFFDT criteria method for Left Outer Joins (putting the join on the Left Outer Join clause), many older queries use the old "or null" method. If you do a modification or "Save-As" of one of these queries you should update the criteria to the new method.

Error 05: Required Prompt on the Where Clause of a Left Outer Join

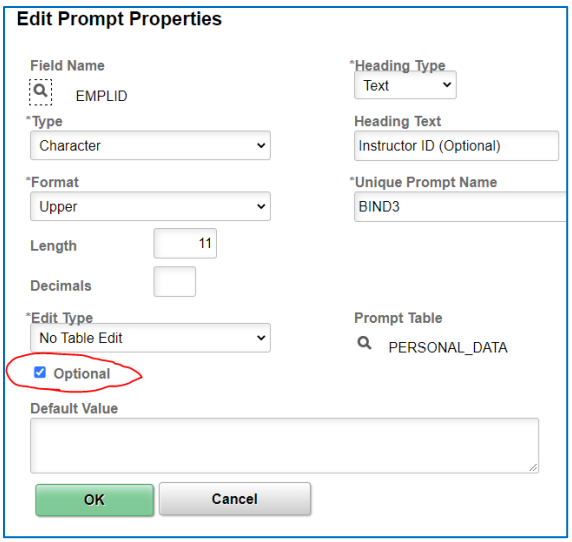
Next I wanted to add a prompt for the instructor EMPLID, so that I could run the query to only show results for a particular instructor.

Just like with the EFFDT criteria. If I add the criteria for this prompt on the where clause, it will restrict all data as if it were inner joined, even though the field the propmt is on (EMPLID) is from a left outer joined record (CLASS_INSTR). Therefore if I make this a required prompt it would be as if the CLASS_INSTR record was inner joined because the results would be restricted to where there is data in the CLASS_INSTR record.

Therefore in this case I should make the prompt optional, so that if the prompt is not used the join acts as a left outer join, but if it is used then it restricts the results like an inner join:



AND	E.EFFDT - Effective Date	Eff Date <=	Current Date	Edit	−	E
AND	E.INSTITUTION - Academic Institution	equal to	:1	Edit	−	E
AND	B.EMPLID - Empl ID	equal to	:3	Edit	−	



This way if I run the query for a specific instructor I will only see the classes that instructor teaches, however I also have the option to leave this prompt blank to see the classes of all instructors (and those without an assigned instructor).

This method of doing prompt criteria can seem counterintuitive. Therefore this is another example of why it is important to know your data and the results you are expecting to get, and to throughly test your queries with your end users and subject matter experts to make sure they are producing the correct data.

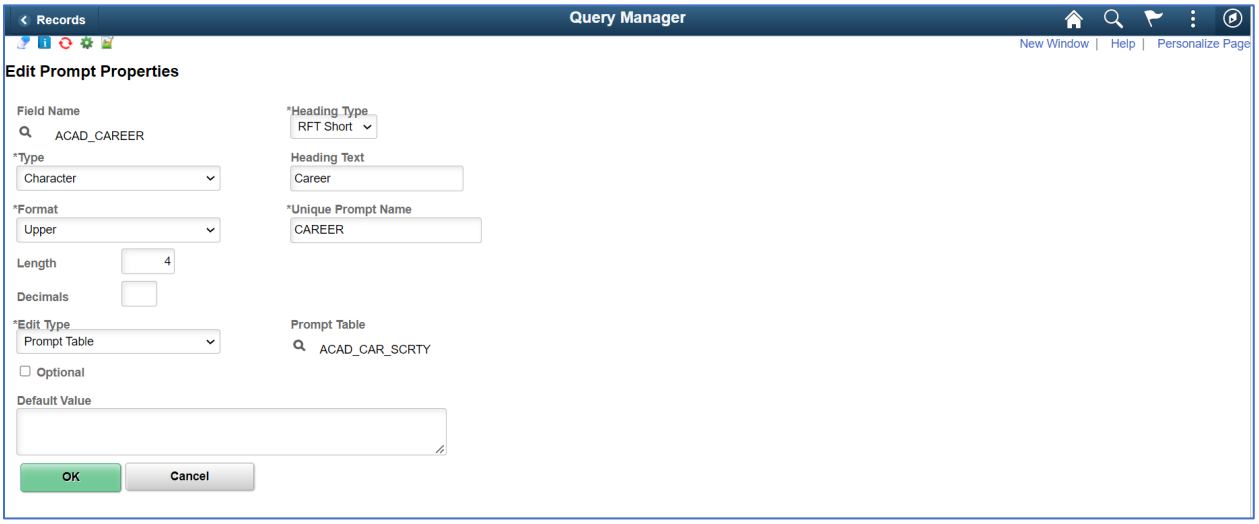
Query Criteria: Hard-Coding vs. Prompts

After completing your ctcLink query training, you are probably already familiar with certain fields that should never be hard-coded in query criteria, such as Institution. Instead of hard-coding Institution (or SetID or Business Unit, etc.) in the query criteria, we require using a secure Institution prompt.

It is common for the ctcLink Reporting Team to see query migration requests for queries that have criteria hard-coded that would be better as prompts to make the queries easier to be used globally. Examples are below.

Academic Career

The Academic Career (ACAD_CAREER) field is commonly hard-coded in ctcLink query criteria. Rather than hard-coding a query with a criterion of ACAD_CAREER = UGRD or ACAD_CAREER = CNED, consider adding a secure ACAD_CAREER prompt instead (see screenshot below).



Make sure you do not enter a value in the Default Value section, as this will override the ACAD_CAREER security.

Example query: QCS_CM_COURSE_SUCCESS

Student Group

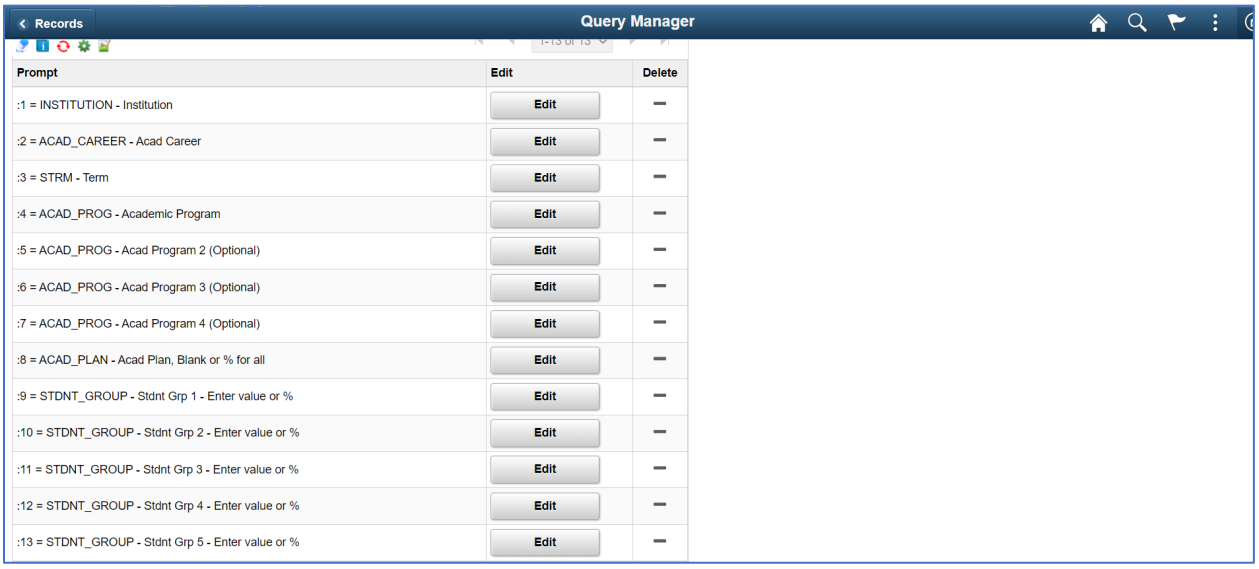
Another field that the ctcLink Reporting Team commonly sees query migration requests submitted with hard-coded values is the Student Group (STDNT_GROUP) field.

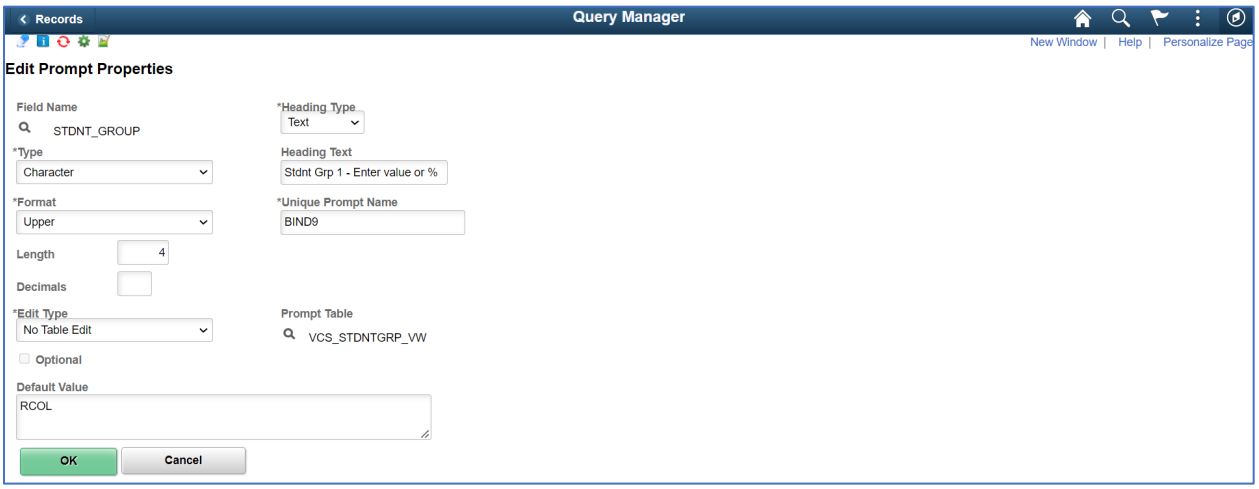
All queries containing Student Group data are required to have Student Group security either by including a secure Student Group prompt(s) or a secure Student Group record (or both). This is because not everyone has access to all Student Group data, and end users should only see Student Group data to which they have access.

Keep in mind that if you hard-code a query with Student Group criteria that are specific to your college, then that essentially makes it a college-specific query because users from other colleges may try to run that query but will not get results.

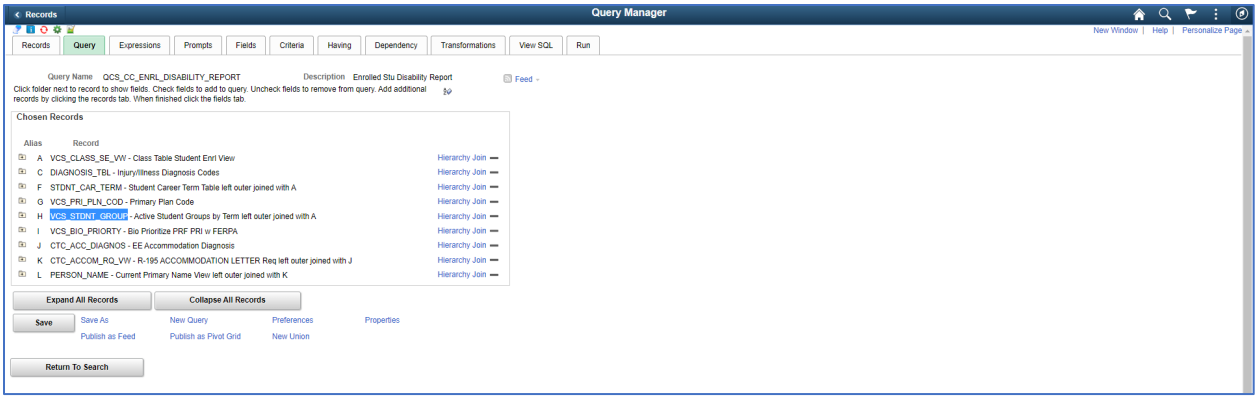
Also keep in mind that if you hard-code a query with global Student Group (i.e., Student Groups that exist at all colleges) criteria (e.g., STDNT_GROUP = SDOC), then end users who don't have access to see data for that Student Group(s) may try to run the query but will not see results returned.

For these reasons, it is better to use a secure STDNT_GROUP prompt(s) instead (see screenshots below).





If you are entering a default value like the example in the screenshot above, then make sure that your query also contains a secure Student Group record so as to not override Student Group security (see screenshot below).

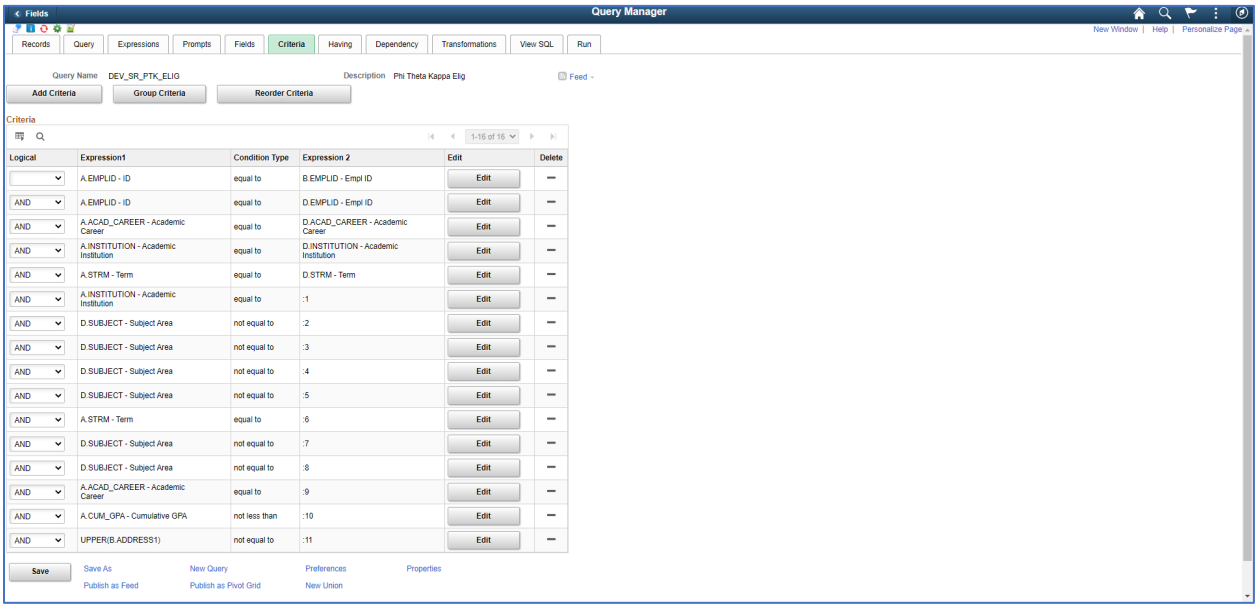


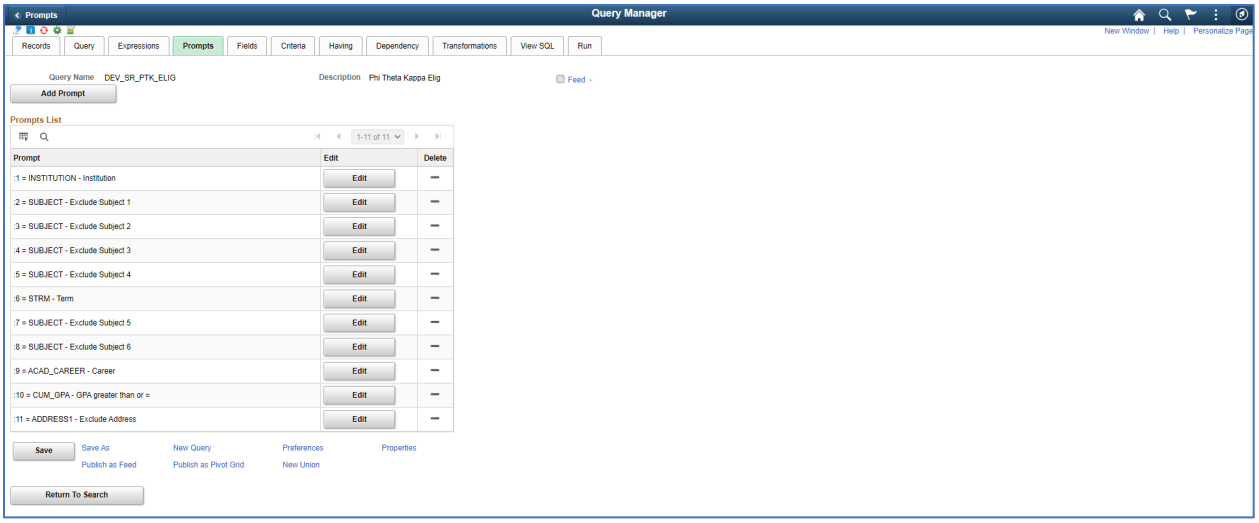
Example query: QCS_CC_ENRL_DISABILITY_REPORT

GPA

Another field that the ctcLink Reporting Team commonly sees query migration requests submitted with hard-coded values is GPA.

If, for example, you need to create a query that only returns students with a cumulative GPA of 3.5 and above, then consider having that criterion as a prompt rather than hard-coded (see screenshots below).





The example query (DEV_SR_PTK_ELIG) referenced in the screenshots above was developed to determine eligibility for an honor society, which requires a cumulative GPA of 3.5 or above. A reason to have this be a prompt with a default value of 3.5 rather than a hard-coded criterion is because other colleges might have different minimum cumulative GPA requirements for eligibility for their chapter of that honor society. Having this prompt makes this query more flexible to be used globally.

Summary

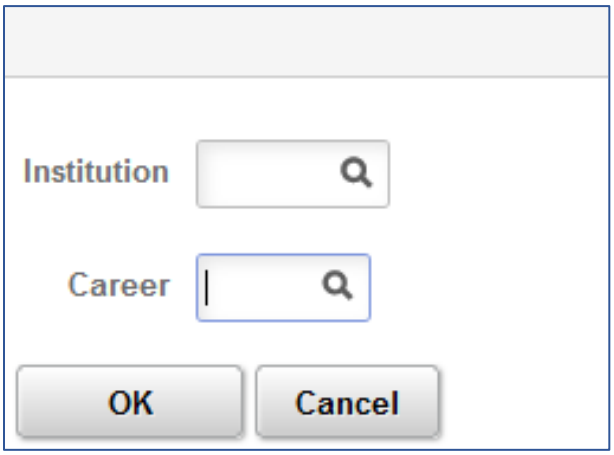
Just about any ctcLink query criteria that are hard-coded can be made into prompts instead. While it may be tempting to include college-specific criteria in a ctcLink query, doing so makes it more difficult for the query to be used globally. If you find yourself adding college-specific criteria to a ctcLink query, please consider using dataLink to create the query instead.

If hard-coding ctcLink query criteria absolutely ***has to*** be done, then be sure to include details about why you chose to hard-code in the query definition. Keep in mind that hard-coded ctcLink queries you create will be found and used by other colleges, and they may even be copied and used as the basis for other queries. That means that if the original query contains criteria that are hard-coded, then copies of that query will also contain criteria that are hard-coded unless those criteria are changed.

Related Prompts

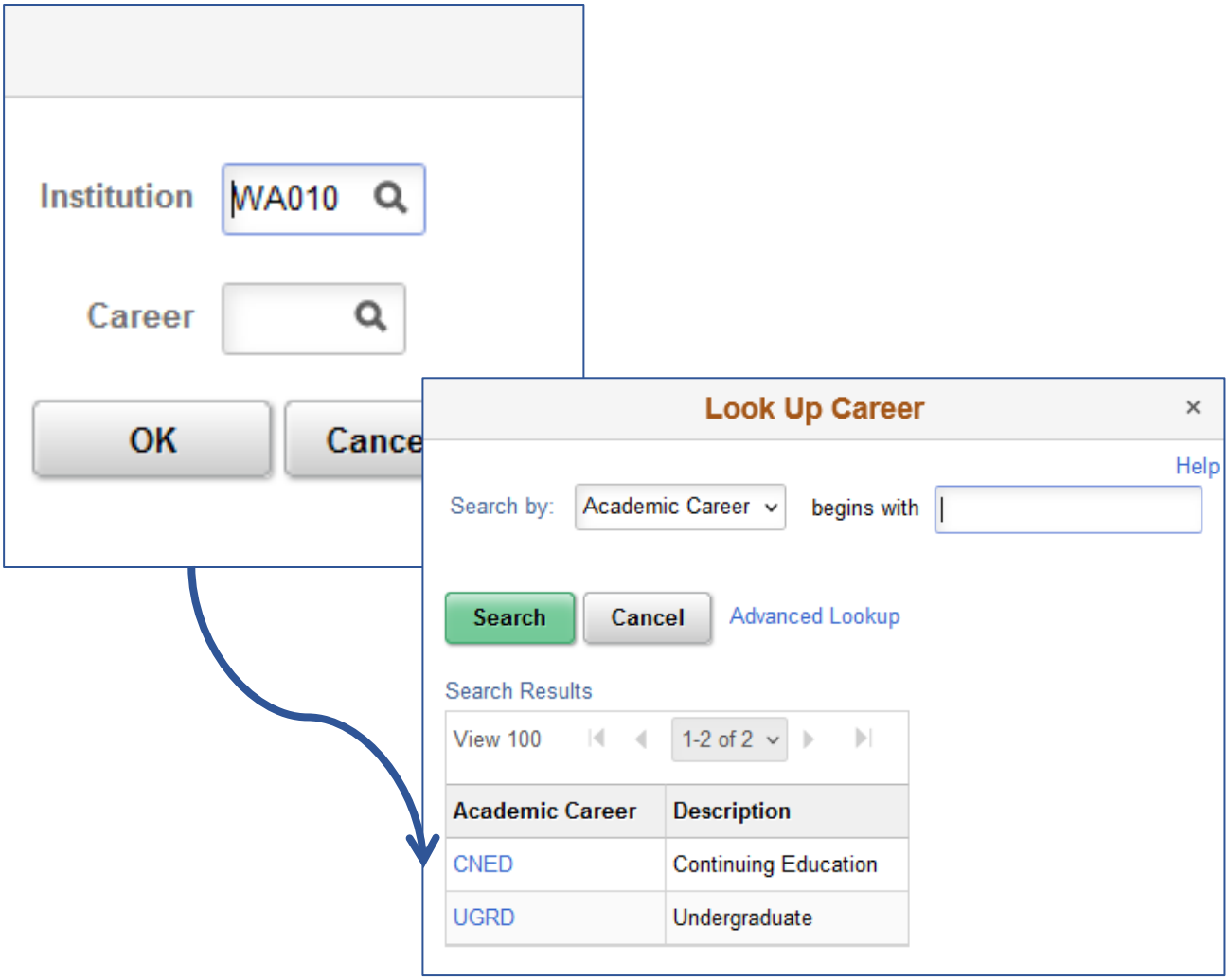
Related prompts are prompts which are dependent on the value from another prompt. ACAD_CAREER and DEPTID are good examples of related prompts. ACAD_CAREER is related to (dependent on) INSTITUTION and DEPTID is related to SETID.

The order of prompt entry can be very important because of this. We recommend to always include the college identifying prompt as the first prompt to avoid any potential difficulties. In this example, I left the INSTITUTION prompt blank and tried to look up a value for ACAD_CAREER first.



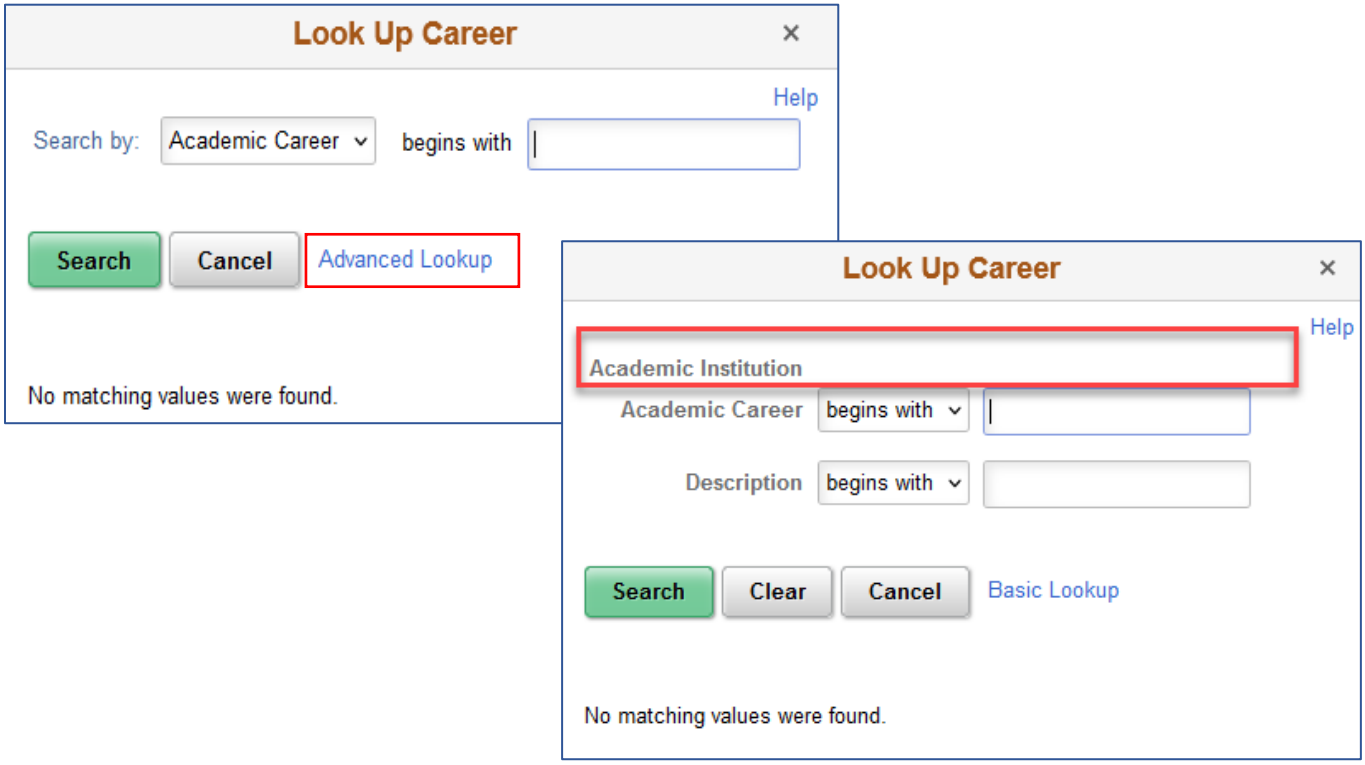
Instead of a list of values I see the error message “No matching values were found”

When I go back and enter in a valid value in the Institution prompt and then try to look up a value in the academic career prompt, I see the list of values I was expecting.



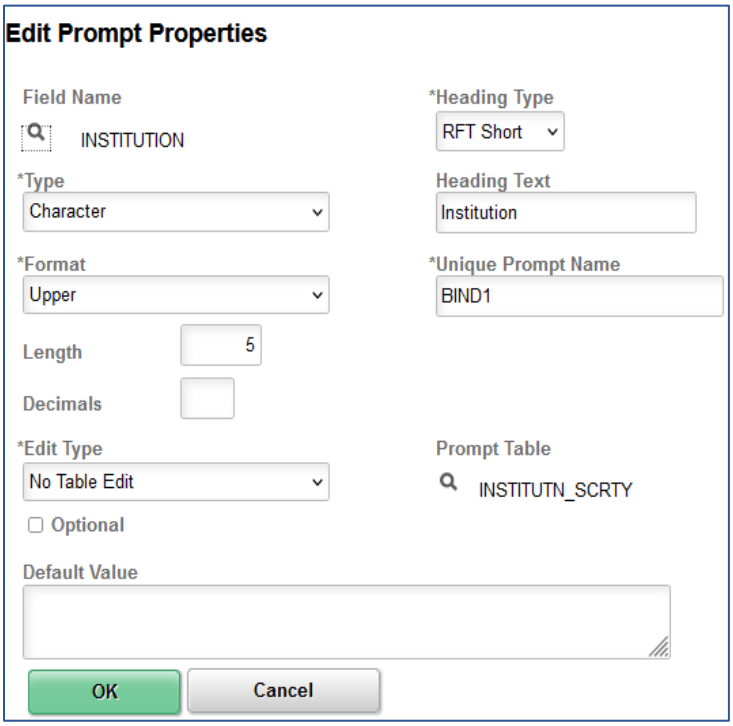
How To Know if a Prompt is Related and What Field it is Related To

The document Query Prompt Tables to Use by Pillar lists out known related prompts. If the prompt is not indicated as related in this document then the best indication of a related prompt (though it could be an incorrect prompt table) is when there are no options when using the magnifying glass to look up prompt values. Click on "Advanced Lookup" and you will see a field with no selection box next to it. That is the field that the prompt is related to which will need to be added as a prompt *before* the related prompt. Also, please send a message to dataservices@sbctc.edu if you find a related prompt we do not have listed.



Which Prompt Table Edit Type to Use

Determining the edit type depends on the desired effect for the end user when entering a value into the prompt.



Translate Table and Yes/No Table edit types will default in automatically based on the field selected for the prompt. No Table Edit and Prompt Table edit types are selected by the developer. When in doubt, choose Prompt Table edit type.

EDIT TYPE	'PROMPT TABLE FIELD' FUNCTION	EFFECT ON THE PROMPT

NO TABLE EDIT	• A record is not required to be added to the 'Prompt Table Field'.
	  • If added, the values in the record provide a look up list and a comparison, but not validation, to the prompt entry.
	  • Does not prevent invalid prompt entry values.
		• Look up list available only if a record is entered in the 'Prompt Table Field'.
		• No validation of the prompt entry made by the end user.
		• Invalid entries are not prevented, but the query will not return results with an invalid prompt entry.

PROMPT TABLE	• A record is required to be added to the 'Prompt Table Field'.
	  • Validates the prompt entry against the values in the record.
	  • Prevents invalid prompt entry values.
	  • The values in the record provide users a look up list.
		• Look up list will always be available.
		• The end user's prompt entry is validated against the values in the record added to the 'Prompt Table Field'.
		• The query will not run with an invalid value.

TRANSLATE TABLE	• None. Not used for this edit type.	• Used on fields with a translate value. • Provides a drop down selection list of predefined values in the prompt.
YES/NO TABLE	• None Not used for this edit type.	• Creates a 'Yes' or 'No' prompt checkbox. Checked = 'Yes', Unchecked = 'No'.

Figure 1 Edit Type/Prompt Table Field Function Crosswalk

Live Example of No Table Edit and Prompt Table Edit Types

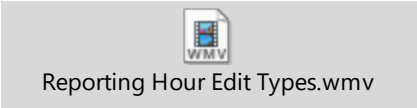


Figure 2 Edit Type Gif
Double Click to open

The video in the link above shows in detail the No Table Edit and Prompt Table edit types. In this query Institution has been set up using a Prompt Table Edit Type while the Academic Program prompt has been set up using an edit type of No Table Edit.

The video shows what happens when I run the query and put in an incorrect institution code. I got a small message saying that no matching values were found but I'll just continue on. When I get to the next required prompt I get an error message window saying that there is an invalid value in the Institution prompt. The prompt box itself turns red and it lets me know that it's not going to run. Even if I try to run it anyway, I can't. It won't let me move forward. To run the query, I have to go back and select a valid institution because the prompt table edit type was used for this prompt.

In contrast the Academic Program prompt used the edit type of No Table Edit so if I put in an incorrect value there, I still get a small message saying that no matching values were found. This is because I put a record in the Prompt Table field in the Edit Prompt Properties page and it will compare the value entered into the prompt against the values in the record added to the Prompt Table field. It is a comparison only, not a validation as nothing will stop me if I continue past the error message. When I try to run the query it will run. I just won't get any results because it tried to filter it on a value that doesn't exist.

Translate Values and the PSXLATITEM table

The same way that fields with a translate value don't need a record in the Prompt Table field, they also don't require a join on the PSXLATITEM table. Check the fields tab. If there is a value of either N, L or S in the XLAT column then the field has a translate value so any prompt will default to a Translate Table type and does not require a record in the Prompt Table field in the Edit Prompt Properties page.

Fields

Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text
1	A.INSTITUTION - Academic Institution	Char5				Institution
2	A.STDNT_ENRL_STATUS - Student Enrollment Status	Char2		N		Status

Which Prompt Table to Use in the Prompt Table Field

The document Query Prompt Tables to Use by Pillar lists which table to use in the Prompt Table field in the Edit Prompt Properties page. Many times, the record listed is a secure record and will determine which rows of data in the record the end user will be able to view.

If using a field as a prompt that is not listed in the document, choosing a record ending in _TBL is a good place to start looking.

Campus Solutions Pillar (CS) Prompt Tables

Required Secure CS Prompt Tables

If one of the following fields is used in a CS Query prompt, the following prompt tables must be used.

Field	Prompt Table
INSTITUTION	**INSTITUTN_SCRTY**
BUSINESS UNIT	**SP_BU_SF_OPRVW**
SETID	**SP_SETID_OPRVW**

Standard Recommended CS Prompt Tables

Field	Prompt Table
ACAD_CAREER*/**	ACAD_CAR_SCRTY**/****
ACAD_ORG	ACAD_ORG_SRCH
ACAD_PLAN	ACAD_PLAN_TBL
ACAD_PROG	ACAD_PROG_TBL
ACAD_STNDNG_ACTN	VCS_ACAD_STACTN
ADMIT_TYPE	ADMIT_TYPE_TBL
ADVISOR_ID*	VCS_ADVISOR_PVW*
AID_YEAR (FA)	FED_AID_YR_TBL
CHECKLIST_CD (FA)	CS_CHKLST_TBL
CLASS_NBR	SE_CLASS_NBR_VW
CRSE_ATTR	CRSE_ATTR_TBL
EMPLID (STUDENT ONLY)*	VCS_PERS_SEC_VW*
EXT_ORG_ID	EXT_ORGSCHL_VW
FACILITY_ID	FACILITY_TBL
INSTRUCTOR_ID*	VCS_INSTRUC_PVW*
ITEM_TYPE*	ITEM_TYPE_BU_VW*
ITEM_TYPE (FA)*	ITEM_TYPE_FA_VW*
SESSION_CODE	SESSION_CODE_VW
SRVC_IND_CD	SRVC_IND_CD_TBL
SRVC_IND_REASON	SRVC_IN_RSN_TBL
STDNT_GROUP*/**	VCS_STDNTGRP_VW**/****
STRM	TERM_VAL_TBL
SUBJECT	SUBJECT_TBL

*Related Prompt – must be a child prompt to an Institution, Business Unit, or SETID prompt
**Additional SACR Security required

INSTITUTION vs SETID vs COMPANY vs BUSINESS_UNIT

College identifying fields are the most important to use a secure prompt record with. Each query created requires a college identifying secure prompt. The college identifying prompt requires a secure record in the Prompt Table field in the Edit Prompt Properties page.

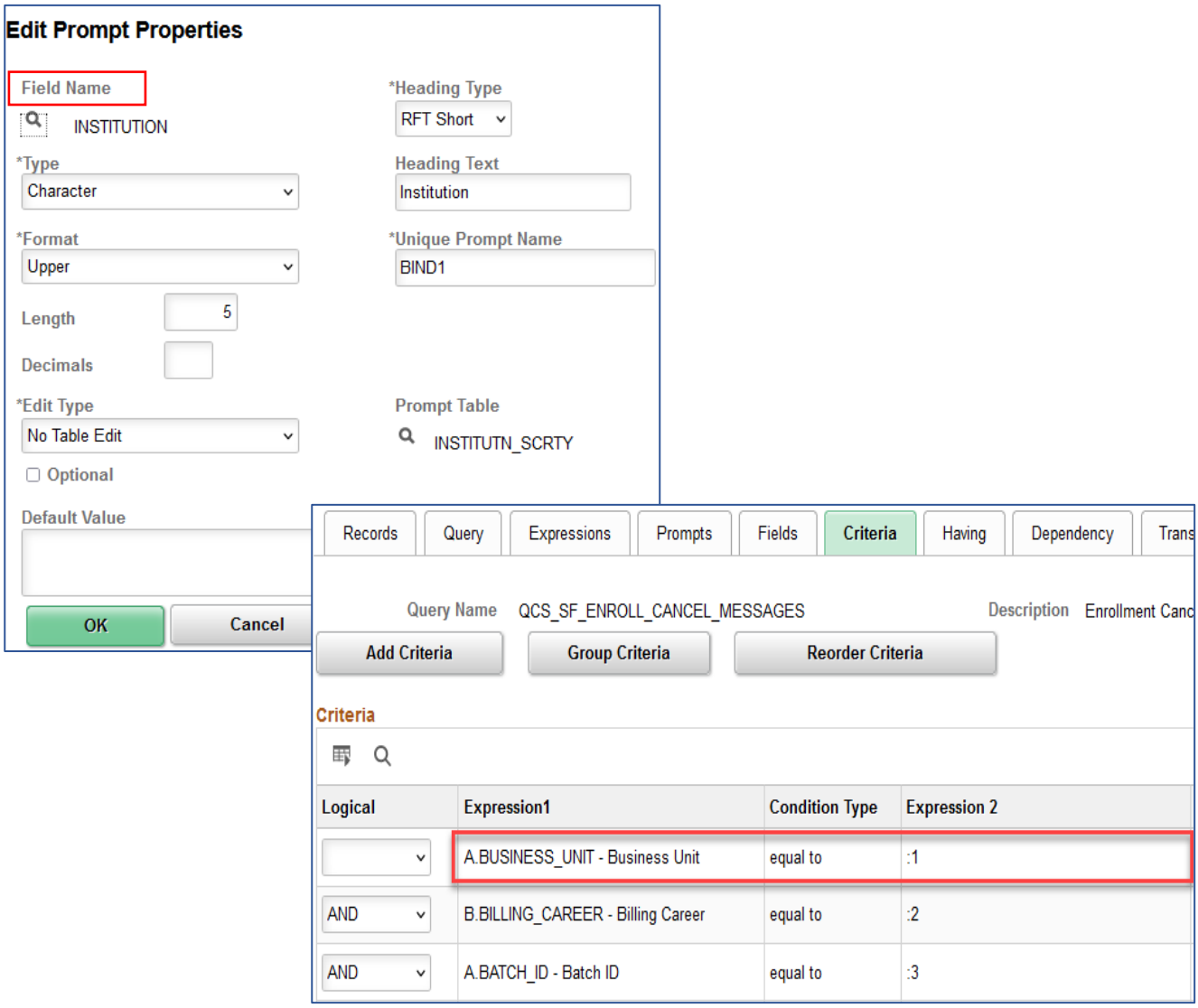
While these fields all identify an institution, they are not all the same and do not always contain the same data.

Creating joins or prompts between any of these fields requires careful consideration of the data. There are times when one table might contain different values than one would think. For example, while it is possible to create a prompt using INSTITUTION as the field in the prompt but have the criteria belong to BUSINESS_UNIT, you will need to first ensure that the data is the same in both fields.

Why Join or Prompt on Different College Identifying Fields

In this query QCS_SF_ENROLL_CANCEL_MESSAGES it works to set up the prompt that way because the data is the same in both the business unit and the institution. But as it's not always the case, check the data first.

Setting it up this way allows the related prompts to work. Remember, ACAD_CAREER is related to INSTITUTION.



Data Differences between College Identifying Tables

The table SF_PAYMENT shows how the INSTITUTION and BUSINESS_UNIT fields can be different even within the same record.

Figure 3 SF_PAYMENT Table INSTITUTION and BUSINESS_UNIT fields

Expressions | Prompts | Fields | Criteria | Having | Dependency | Transformations

d to Excel | Download to XML

Firs

Institution	Unit
	WA171
	WA172
	WA220
WA020	
WA030	
WA040	
WA040	WA040
WA050	
WA050	WA050
WA062	
WA062	WA062
WA063	
WA064	
WA064	WA064
WA070	
WA080	
WA090	
WA100	
WA110	
WA120	
WA130	
WA130	WA130

The JOB and JOBCODE_TBL records also highlight the potential differences between college identifying fields

Figure 5 JOBCODE_TBL SETID and COMPANY fields

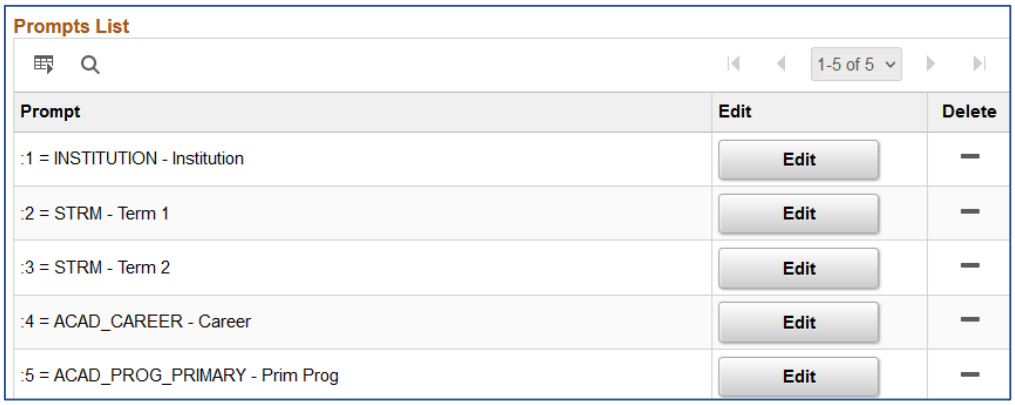
Set ID	Co
WA000	000
WA000	890
WA010	000
WA010	010
WA020	000
WA020	020
WA030	000
WA030	030
WA040	000
WA040	040
WA050	000
WA050	050
WA060	000
WA060	060
WA070	000
WA070	070
WA080	000
WA080	080
WA090	000
WA090	090
WA100	000

Figure 5 JOB Table COMPANY and BUSINESS_UNIT fields

Co	Unit
010	HR010
020	HR020
030	HR030
040	HR040
050	HR050
060	HR060
060	HR062
060	HR063
060	HR064
070	HR070
080	HR080
090	HR090
100	HR100
110	HR110
110	HR111
110	HR112
120	HR120
130	HR130

Floating Prompts

Prompts can be created through the Add Criteria icon or through the Prompts tab. If they are created using the Prompts tab they must be manually added as criteria in the query. Here we see a prompt on ACAD_PROG_PRIMARY in the list of prompts.

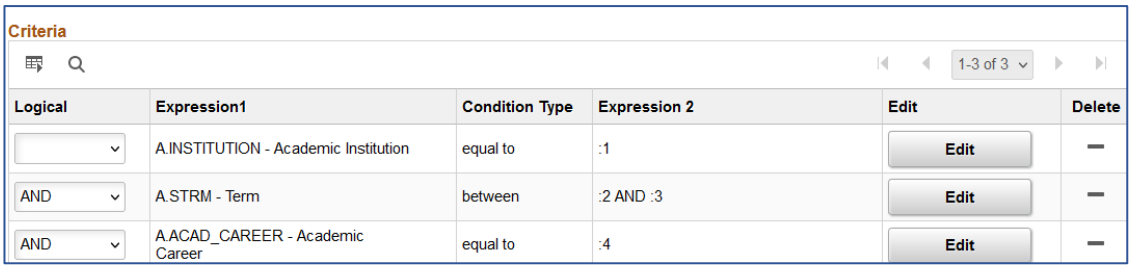


Prompts List

1-5 of 5

Prompt	Edit	Delete
:1 = INSTITUTION - Institution	Edit	
:2 = STRM - Term 1	Edit	
:3 = STRM - Term 2	Edit	
:4 = ACAD_CAREER - Career	Edit	
:5 = ACAD_PROG_PRIMARY - Prim Prog	Edit	

However, it is not listed as a criterion on the Query tab. It is "floating". It is not filtering the query results. Either tie the prompt as criteria or delete it from the Prompts tab.



Criteria

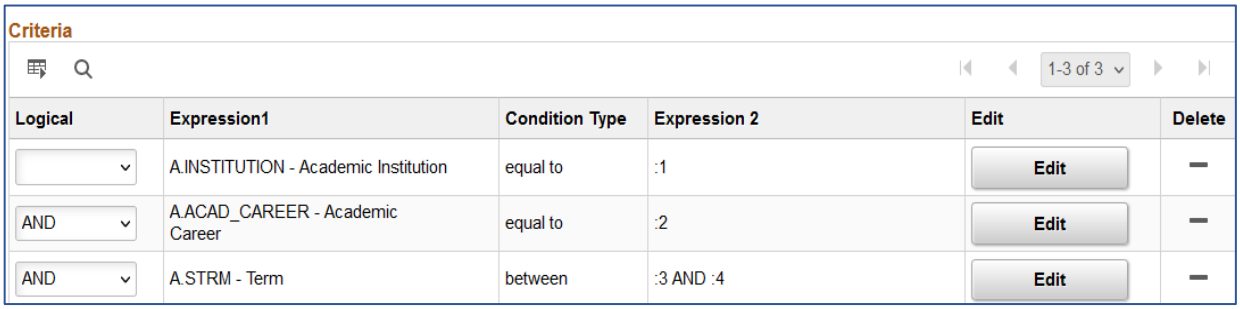
1-3 of 3

Logical	Expression1	Condition Type	Expression 2	Edit	Delete
	A.INSTITUTION - Academic Institution	equal to	:1	Edit	
AND	A.STRM - Term	between	:2 AND :3	Edit	
AND	A.ACAD_CAREER - Academic Career	equal to	:4	Edit	

Prompts Used in Expressions Require Manual Clean-Up

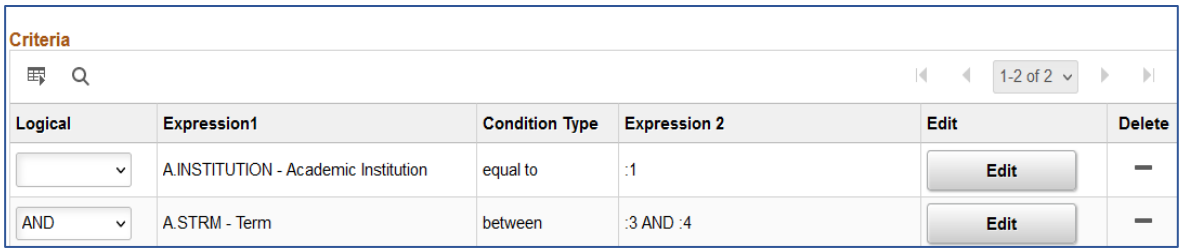
When a prompt is deleted, the remaining prompts will re-number. But prompts used in an expression will require manual clean up.

Here we see prompts :1 - :4.

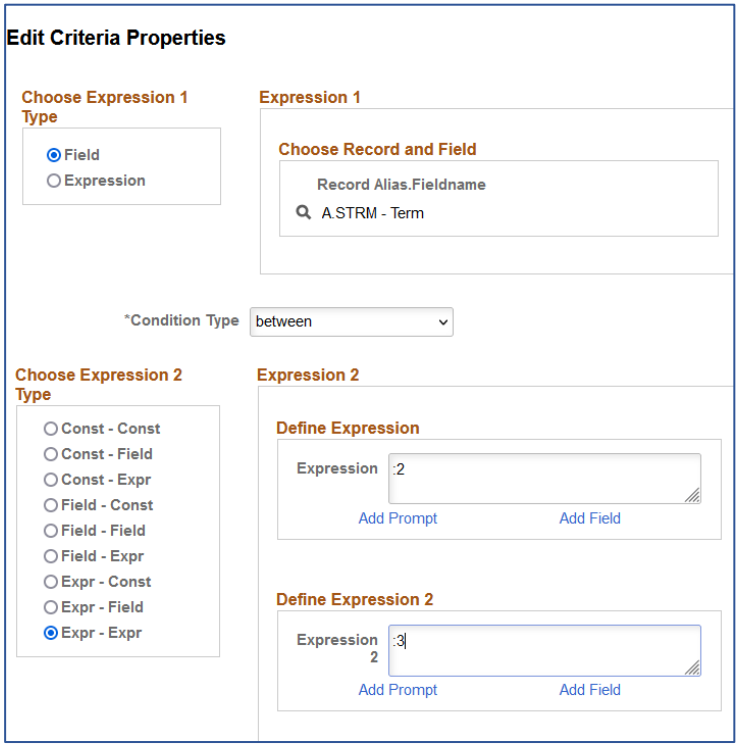


When I delete the prompt on ACAD_CAREER, the term prompts, which were :3 and :4 now move to :2 and :3.

But the criteria on TERM that used on expression still is referring to the term prompts as :3 and :4



I have to go into the criteria and edit the Expression 2 expressions to reflect the new prompt numbers which are :2 and :3.

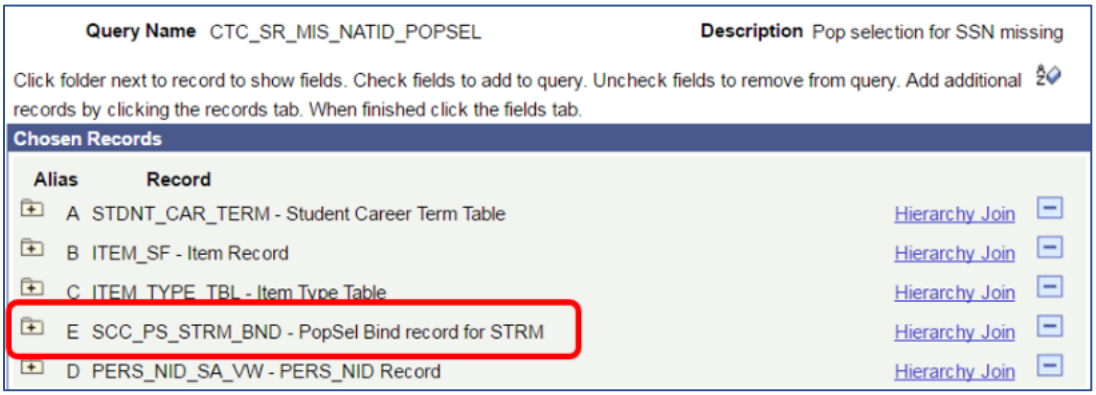


Pop Select Queries

A Pop Select query is just a tool to gather the IDs. It was never meant to be a standalone query that provides data outside of what is needed to run the batch process it is associated with.

For a query to be a pop select query it must contain a bind record. Each batch process that allows pop selects has an associated bind record. The bind record contains all the fields necessary to run that batch process.

- If a query uses a bind record, it must be associated to a process and be saved in the POPULATION SELECTION folder.
- A stand-alone query may not use a bind record. Queries using bind records are not captured as part of the annual query maintenance as they are associated to a business process so bind records should not be used for data queries.



For the full explantion of using bind records, please see this email from Ivy Brent to the querydev listserv dated 10/27/2022

Ctclink_qrydev Pop Select Queries.msg

Figure 6

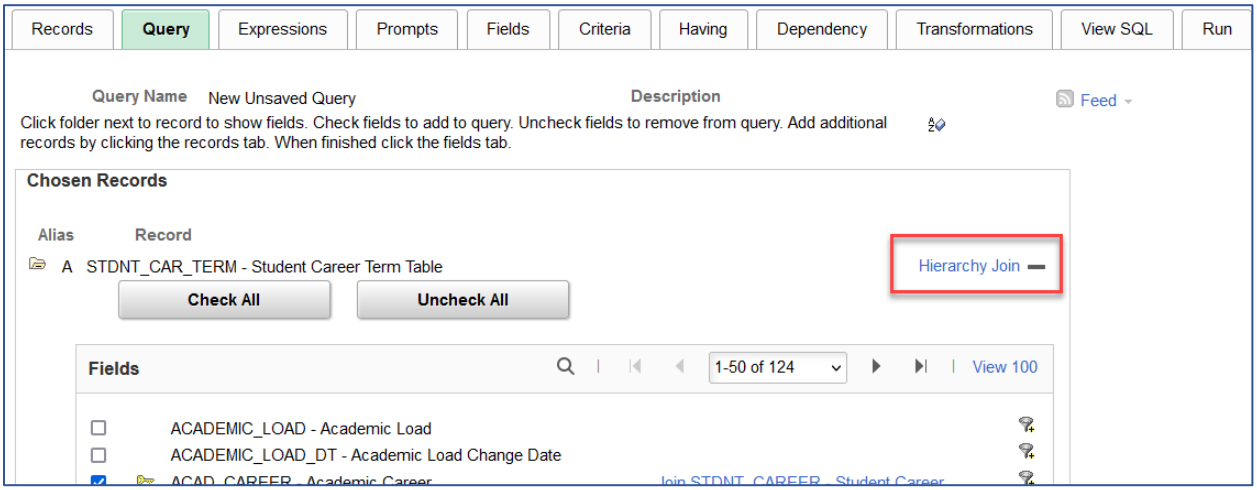
What Join Method to Use

There are three methods for creating a join.

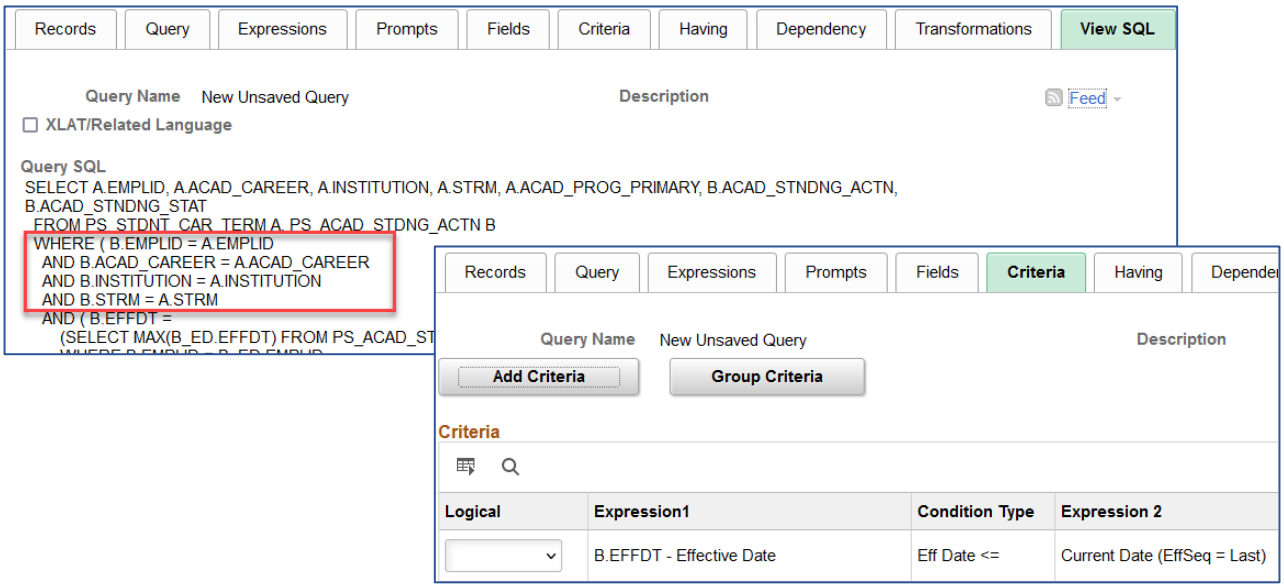
1. Hierarchy Join
   - Do Not Use
2. Pre-defined Join
   - Do Not Use
3. Any Join
   - Recommended

Hierarchy Join

- Not allowed for migration.

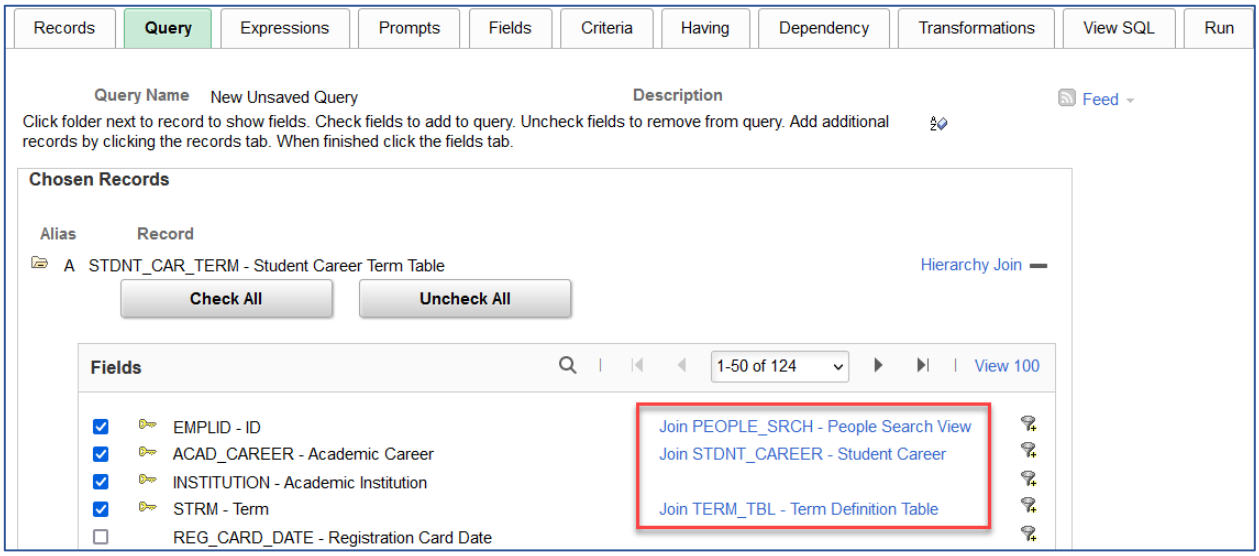


This join method only allows for standard joins between records and while the join criteria will display in the SQL code, it is not available for the Query Developer to manipulate in the Criteria tab.

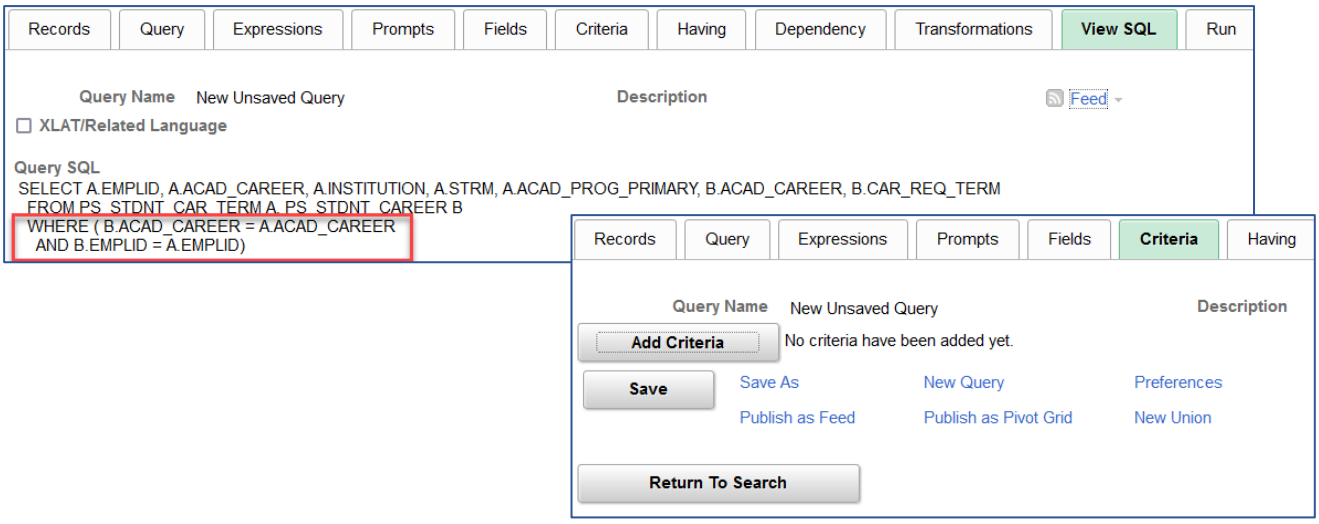


Pre-defined Join

- Not allowed for migration.

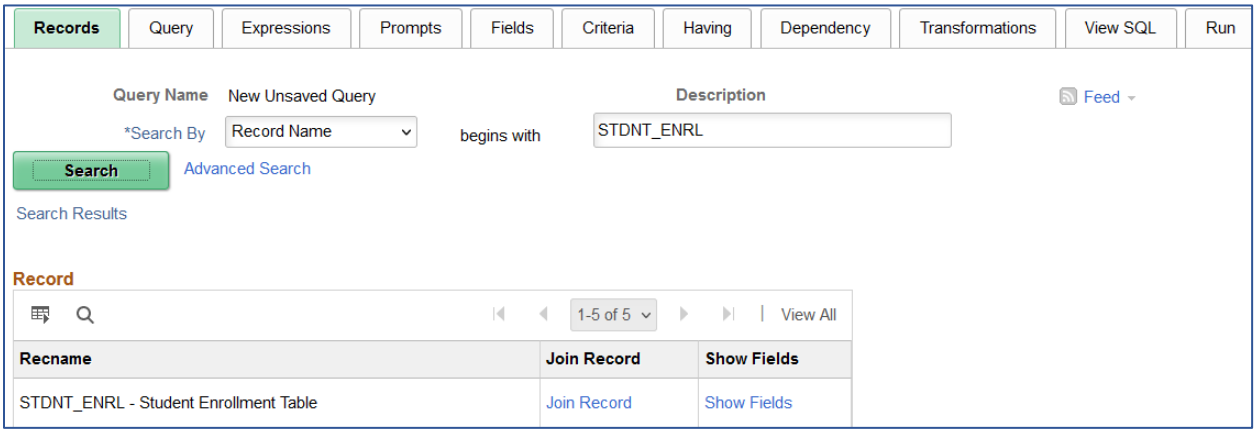


This join method allows for both left outer and standard joins between records. But again, while the join criteria will display in the SQL code, it is not available for the Query Developer to manipulate in the Criteria tab.

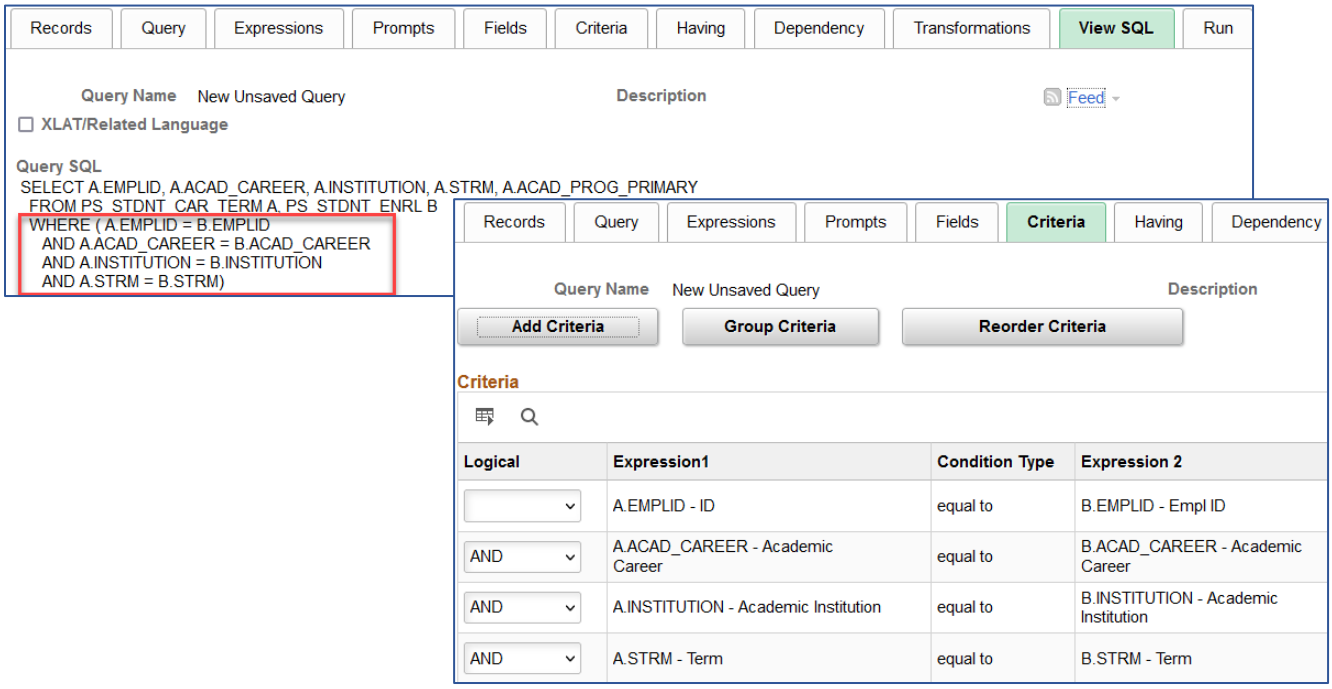


Any Join

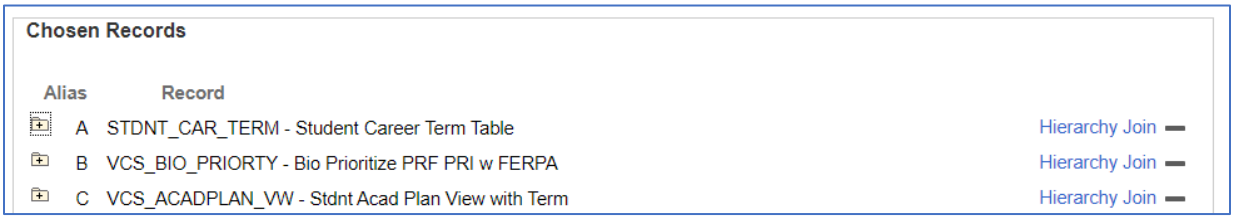
- Recommended.



This join method allows for both left outer and standard joins between records. The join criteria will display in the SQL code and it is also available for the Query Developer to manipulate in the Criteria tab.



Redundant Records



Example:

VCS_ACADPLAN_VW joins to the STDNT_CAR_NBR detail in STDNT_CAR_TERM in the view.

If another field is needed from STDNT_CAR_TERM, performance will suffer if I add STDNT_CAR_TERM to the query. Ideally, I would submit a ticket asking to add an additional field to the custom view, preventing the need of the redundant STDNT_CAR_TERM record.

Custom Records

Need another field in a custom record or must join a seemingly redundant record?

Submit a ticket to ask Data Services to add the field!

Delivered Records

I take the time to research if there is a delivered record or view that may have the field(s).

I use metaLink – search by Field(s) Name

https://dataservicesmetalink.sbctc.edu/QRSearch

If you are logged in, you will see Records/View definition returned as well as reporting objects

What Now?

I try not to leave both records in the query. I will remove the record that may no longer need to be included because the replacement record includes the fields I was using the original record for.

This may require rewriting the query. Worth it!

Effective Dating

One of the most important things I feel that a query developer can understand is Effective Dating when joining records that have time-based fields to consider.

This concept is relevant in every Pillar, joining Effective Dated records to time-based records, such as:

- Term in CS Pillar
- Term End Date (most common)
- Term Begin Date (important to be used between Course and Class)
- Pay End Date in HCM Pillar
- Fiscal Year End Date in FIN Pillar

Normally, we want to retrieve data from the Maximum Effective Dated row that is less than or equal to the Date related in the other time-based record we are joining to.

Think about the relation of the data in the record to the term/pay end date/fiscal year, etc. and its importance, validity, or accuracy on what you are joining to.

There may be times you may want the most recent or last effective dated row:

- Descriptions
- Translate Values (PSXLATITEM)

One of the best ways to understand effective dating and the impact of the join criteria is to really know your data. Here are a few ways I "get to know my data".

- I select all fields in a table or view for a single ID and verify my understanding of the fields/values.
- I will remove the effective dating logic that defaults into the query so that I can see all rows to help understand how this data may flow over time into this record.
- If I see something or question what other values may be in a field, I create a new query and select where the Field <> to that value to see what others are present, are there blanks, etc.
- I ask for a specific example that I can track through the tables to give me an expectation of data to be returned based on my understanding of the business process. I look at the data to prove the expectation is true or false. This helps me to ensure that I understand the relation of the data in the record and its importance, validity, or accuracy based on the keys fields and the fields I am joining to.
- I will create different queries to compare and see the difference based on different join criteria or effective date logic.
- I will compare counts, compare data, review differences with the institutional subject matter

expert.

- I thoroughly test the query in PCD
- I work to understand the Business Process
- Find related QRGs
- Work closely with Subject Matter Experts
- Review previously developed queries and reports
- This is last on purpose because what works for one doesn't work for all, query logic needs to be reviewed for accuracy and validity for the specific need/institution/business process, etc.
- I work closely with Subject Matter Experts to test the query results *before* migrating to Production.

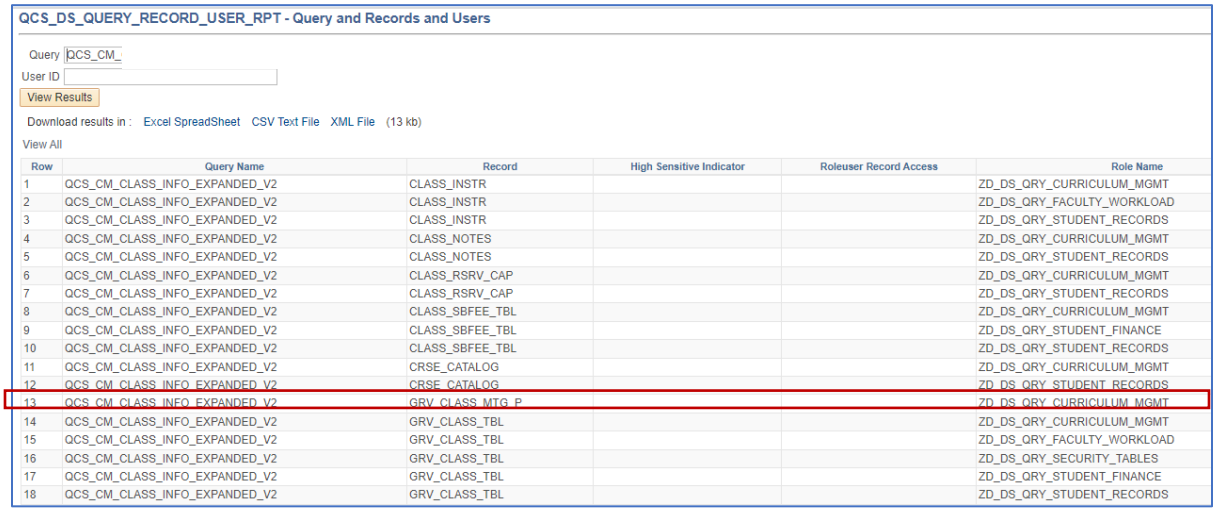
There are so many tips and tricks related to "Getting to know your data" that we plan to have an entire Reporting Hour dedicated to this in the future.

QUERY SECURITY

When adding a new table to an existing query run QXX_DS_QUERY_RECORD_USER_RPT to determine if additional roles will be needed to run the query or if the record should be added to an existing access group/role to ensure users with access will continue to have access without having to add additional query security roles.

Submit a ticket to ask Data Service to add record to an access group/role

In the example below, the record GRV_CLASS_MTG_P was added and is only accessible via the ZD_DS_QRY_CURRICULUM_MGMT role. However, all other records in this query are also accessible with the ZD_DS_QRY_STUDENT_RECORDS role.



QCS_DS_QUERY_RECORD_USER_RPT - Query and Records and Users

Query QCS_CM_

User ID

View Results

Download results in : Excel SpreadSheet CSV Text File XML File (13 kb)

View All

Row	Query Name	Record	High Sensitive Indicator	Roleuser Record Access	Role Name
1	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_INSTR			ZD_DS_QRY_CURRICULUM_MGMT
2	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_INSTR			ZD_DS_QRY_FACULTY_WORKLOAD
3	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_INSTR			ZD_DS_QRY_STUDENT_RECORDS
4	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_NOTES			ZD_DS_QRY_CURRICULUM_MGMT
5	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_NOTES			ZD_DS_QRY_STUDENT_RECORDS
6	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_RSRV_CAP			ZD_DS_QRY_CURRICULUM_MGMT
7	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_RSRV_CAP			ZD_DS_QRY_STUDENT_RECORDS
8	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_SBFEE_TBL			ZD_DS_QRY_CURRICULUM_MGMT
9	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_SBFEE_TBL			ZD_DS_QRY_STUDENT_FINANCE
10	QCS_CM_CLASS_INFO_EXPANDED_V2	CLASS_SBFEE_TBL			ZD_DS_QRY_STUDENT_RECORDS
11	QCS_CM_CLASS_INFO_EXPANDED_V2	CRSE_CATALOG			ZD_DS_QRY_CURRICULUM_MGMT
12	QCS_CM_CLASS_INFO_EXPANDED_V2	CRSE_CATALOG			ZD_DS_QRY_STUDENT_RECORDS
13	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_MTG_P			ZD_DS_QRY_CURRICULUM_MGMT
14	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_TBL			ZD_DS_QRY_CURRICULUM_MGMT
15	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_TBL			ZD_DS_QRY_FACULTY_WORKLOAD
16	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_TBL			ZD_DS_QRY_SECURITY_TABLES
17	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_TBL			ZD_DS_QRY_STUDENT_FINANCE
18	QCS_CM_CLASS_INFO_EXPANDED_V2	GRV_CLASS_TBL			ZD_DS_QRY_STUDENT_RECORDS

How do you know when secure tables need to be added and what tables to use?

Get a better understanding of Query Security by reviewing the Understanding Query Security training

https://www.sbctc.edu/resources/documents/colleges-staff/data-services/peoplesoft-ctclink/ps-query-102-query-security-training-manual.pdf

Review our Secure Prompts document

https://www.sbctc.edu/resources/documents/colleges-staff/data-services/peoplesoft-ctclink/prompt-tables-to-use.pdf

CS is the pillar where these decisions most often need to be made

In HCM/FIN, this decision more often is when to decide not to use a Highly Sensitive version of a table if those fields are not required in the output

Review the SQL tab from Query Manager to review any OPRID = logic

This indicates there is security being applied on a record and field in the query

SELECT DISTINCT A…

FROM ((((((PS_STDNT_CAR_TERM A LEFT OUTER JOIN (VCS_STDNT_RS D INNER JOIN GRV_QSEC_INSTIT D1 ON (D.INSTITUTION = D1.INSTITUTION AND D1.OPRID = 'CTC_TMORRILL')) ON A.EMPLID = D.EMPLID AND A.ACAD_CAREER = D.ACAD_CAREER AND A.INSTITUTION = D.INSTITUTION AND A.STRM = D.STRM LEFT OUTER JOIN (PS_VCS_FIRST_TERM F INNER JOIN GRV_QSEC_INSTIT F1 ON (F.INSTITUTION = F1.INSTITUTION AND F1.OPRID = 'CTC_TMORRILL')) ON A.EMPLID = F.EMPLID AND A.ACAD_CAREER = F.ACAD_CAREER AND A.INSTITUTION = F.INSTITUTION AND F.STDNT_CAR_NBR = A.STDNT_CAR_NBR) LEFT OUTER JOIN (PS_VCS_LAST_TERM G INNER JOIN GRV_QSEC_INSTIT G1 ON (G.INSTITUTION = G1.INSTITUTION AND G1.OPRID = 'CTC_TMORRILL')) ON A.EMPLID = G.EMPLID AND A.ACAD_CAREER = G.ACAD_CAREER AND A.INSTITUTION = G.INSTITUTION AND G.STDNT_CAR_NBR = A.STDNT_CAR_NBR) LEFT OUTER JOIN (VCS_STDNT_RACE H INNER JOIN GRV_QSEC_INSTIT H1 ON (H.INSTITUTION = H1.INSTITUTION AND H1.OPRID = 'CTC_TMORRILL')) ON A.EMPLID = H.EMPLID AND A.INSTITUTION = H.INSTITUTION AND A.STRM = H.STRM AND H.PRIMARY_INDICATOR = 'Y') LEFT OUTER JOIN PS_ACAD_DEGR I ON A.EMPLID = I.EMPLID AND A.ACAD_CAREER = I.ACAD_CAREER AND A.INSTITUTION = I.INSTITUTION) LEFT OUTER JOIN PS_ACAD_DEGR_PLAN J ON I.EMPLID = J.EMPLID AND I.STDNT_DEGR = J.STDNT_DEGR), PS_VCS_BIO_PRIORTY B, PS_VCS_ACADPLAN_VW C, VCS_STDNT_GROUP E, GRV_QSEC_INSTIT C1, PS_SCRTY_TBL_STGP E1
WHERE (C.INSTITUTION = C1.INSTITUTION
AND C1.OPRID = 'CTC_TMORRILL'
AND E.INSTITUTION = E1.INSTITUTION
AND E.STDNT_GROUP = E1.STDNT_GROUP
AND E1.OPRID = 'CTC_TMORRILL'
AND (A.INSTITUTION = :1
AND A.STRM = :2
AND A.EMPLID = B.EMPLID
AND A.EMPLID = C.EMPLID…

Does your query have an optional or default prompt on a secure prompt field?

If so, then you will need to ensure there is a secure GRV record joined in the query.

Review the SQL to ensure that the OPRID and record field is secure by operator

Best Practice is to have at least one inner joined GRV version of a record in place of the delivered version, this ensures secure data at the Institution level.

Example: Choose GRV_STDNT_ENRL in place of STDNT_ENRL

There are so many things to relay about Query Security that we plan to have an entire Reporting Hour dedicated to this in the future.

Return to Table of Contents

Washington State Board for Community and Technical Colleges