Core-CT EPM Query Advanced
HRMS 201

Contents

Joining Tables, Simple Prompts

Scenario: Build a query to audit retirement plan deduction codes in your agency

Skills: Join tables, simple prompts, hard coded criteria

1. Add tables.

Tables required:
A.CTW_EMPLOYEE_VW
B.CTW_EMPL_DEDUCT

2. Join tables.

Field Joins:
EMPLID
EMPL_RCD

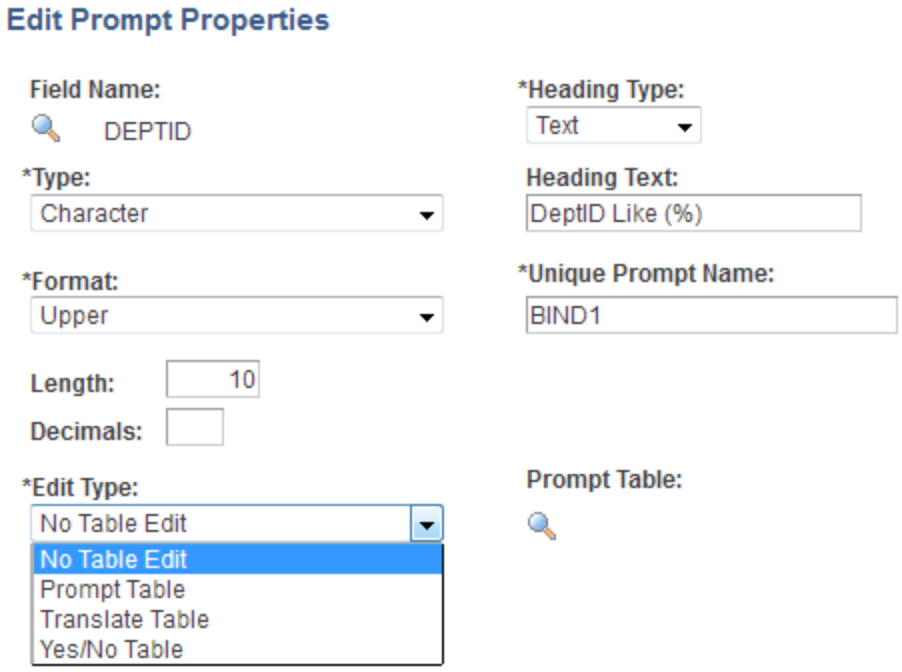
3. Add fields and order the fields to your preference.

Fields required:
A.EMPLID
A.EMPL_RCD
A.NAME
A.BIRTHDATE
A.JOBCODE
A.CT_JOBCODE_DESCR
A.STD_HOURS
A.HIRE_DT
A.UNION_CD
A.JOB_INDICATOR
B.DEDCD
B.CT_DEDCD_DESCR
B.EFFDT

4. Add criteria.

    a. Prompt A.DEPTID like :1

Set edit type prompt to No Table Edit.



b. A.EMPL_STATUS in list (‘A’,’L’,’P’,’S’)
c. B.DEDUCTION_END_DT is null
d. B.DEDCD in list ('RARNSS','RARPSS','RS1PLA','RS1PLC','RS1UB1','RS1UCA','RS1UCC','RS2NSS','RS2SS','RSE1B2','RSE2A','RCPCMR','RPUBDF','RTRNSS','RSTATY','RTRPT','RTRSER','RTRSRP','RTRSSS')

5. Run and test query.

Joining Tables

Scenario: Develop a query quantify OPEB (Retirement Health Fund) deductions. The acronyms OPEB, Other Pension Employee Benefit OTRS, Other Teacher Retirement System.

Note for this class query, we will use the standard version of the employee and detailed payroll tables. For actual determination of the amount an employee has contributed, use the tables:

CTW_DTPY3PCT_VW - Detail Payroll 3Pct View
CTW_EMPL3PCT_VW - Employee 3Pct View

1. Add Tables.

Tables required:
CTW_EMPLOYEE_VW
CTW_DET_PAYROLL

2. Join Tables.

Field Joins
EMPLID
EMPL_RCD
COMPANY

3. Add fields and order the fields to your preference.

Fields required:
A.FULL_PART_TIME
A.ORIG_HIRE_DT
A.REG_TEMP
A.STD_HOURS
B.CHECK_DT
B.CT_AMOUNT
B.CT_TRANSACTION_CD
B.CT_TRANS_CD_DESCR
B.DEPTID
B.EMPLID
B.EMPL_RCD
B.NAME
B.PAYGROUP
B.PAY_END_DT

4. Add criteria.
    a. Prompt B.EMPLID like :1
    b. Prompt B.DEPTID like :2
    c. Prompt B.PAY_END_DT between :3 and :4

d. A.EMPL_STATUS in list (‘A’,’L’,’P’,’S’)
e. A.EFFDT<=B.PAY_END_DT (Effseq=Last)
f. B.PAY_END_DT > June 1, 2023
g. B.CT_TRANSACTION_CD in list (‘OPEB’, ’OPE2’, ’OTRS’ ,’OTR2’, ’ADJOPE’)

5. Run and test query.

Grouping Criteria

Skills: Grouping Criteria, Summary Queries, Aggregate Functions, Having Criteria

- Grouping Criteria
    - The rules of precedence determine the order in which expressions are calculated.

Operator	Meaning
1	Arithmetic operators
2	Concatenation operator
3	Comparison conditions
4	IS(NOT)NULL, LIKE, (NOT) IN
5	(NOT) BETWEEN
6	Not equal to
7	NOT logical condition
8	AND logical condition
9	OR logical condition

    - You can override the default order by using parentheses around the expressions you wish to calculate first. The Query Manager tool designates this functionality as Grouping Criteria.

- Summary Queries

For successful summary queries, display the fewest possible fields. The more fields, the more detail.

- Aggregate Functions

Aggregates are the basic arithmetic functionality provided by the Query Manager Tool, sum, count, maximum, minimum, and average. The application of an aggregate does not limit the return of your query, just changes the display of the data. The aggregate functionality is access via the edit button.

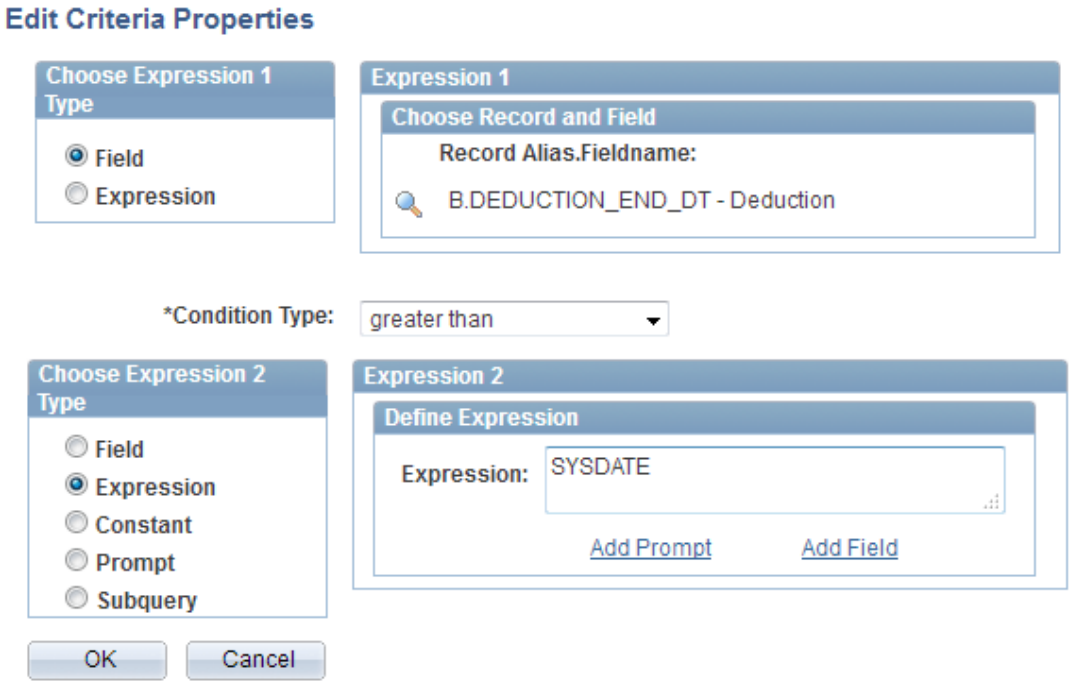
When using aggregates, you will most likely want to change the column heading text. Make sure to click the radio button text.

- Having Criteria

Having criteria is simply criteria established on a field which has an aggregate function applied to it. The criteria can be hard coded or set up as a prompt.

Grouping Criteria Query

1. Locate query created in "Joining Tables, Simple Prompts" section above.
2. Add additional criteria.



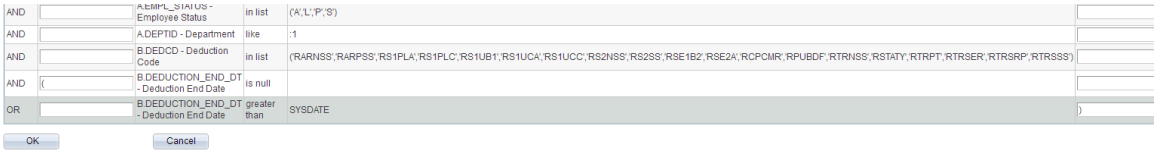
3. If necessary, reorder the criteria so the following criteria are in consecutive rows.

B.DEDUCTION_END_DT is Null
B.DEDUCTION_END_DT greater than SYSDATE

4. Change the logical condition to OR

5. Group the criteria.

From the criteria tab, click the yellow group criteria button and add in parentheses as indicated below.



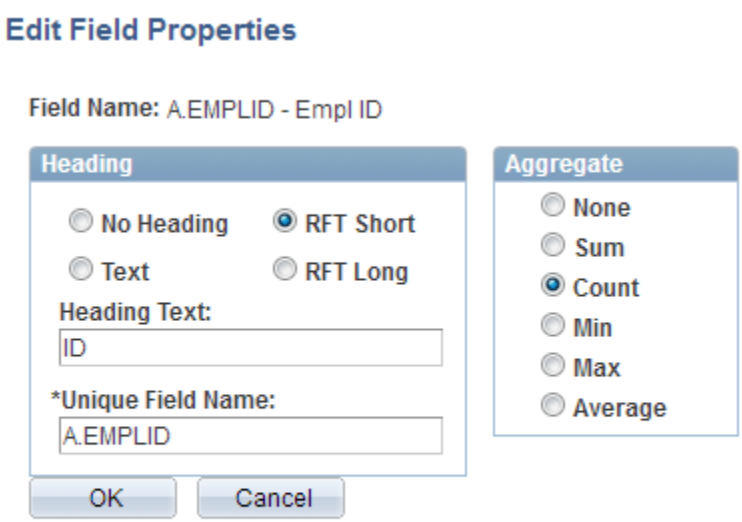
6. Run and test the query.

Note: By grouping the criteria, the order of precedence changes.

Summary query with Aggregate and Having

Scenario: Create a query to count number of active employees per job in selected department.

1. Place the aggregate function of count on the field A. EMPLID



2. The aggregate field COUNT(A.EMPLID) can have criteria placed on it and is displayed on the Having Tab.

COUNT(A.EMPLID) greater than 10

Example public query using aggregates: CT_CORE_HR_AA_WKFRC_ANLYSIS_FT

Expression Query

Skill: Using Expressions

Overview: Expressions are maintained on the Expressions tab in the Query Component and are generally created using Structured Query Language (SQL) syntax and code. Expressions can include calculations, data manipulations or transformations of data and can be used like any other field in that criteria can be placed on the expression. Expressions can also function as prompts and aggregate functions can be applied to them.

PS Query Expressions:

- Concatenate
- Substring
- Decode
- Case
- Date Manipulation

All expressions are established using the Expressions Tab>Add Expression
Expression data type is chosen from the drop down menu. The field type, length and number of decimals are determined by a combination of the field and the syntax it uses and by selecting and typing in the appropriate boxes. SQL syntax for the expression is entered in the Expression Text Box.

To use any expression as a field in your query, click the Use as Field hyperlink after an expression has been created.

Concatenate

Merging or conjoining two or more fields and values into one report column.

Exercise Steps 1

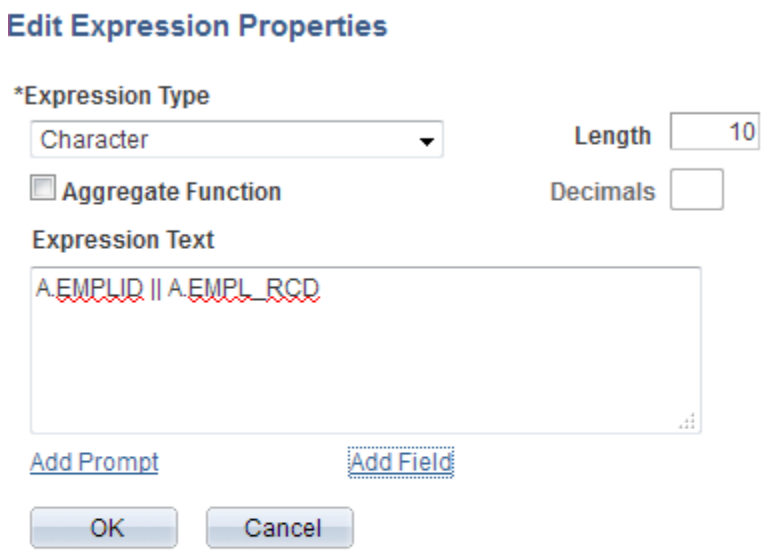
1. Locate query created in “Joining Tables, Simple Prompts” section above.

2. Add Expression:

   Expression Type = Character; Length = 10

Expression Text: A.EMPLID || A.EMPL_RCD

Very Important! Make sure the expression || is preceded and followed by a space. To separate the data from each field concatenated, you can either use a (space) or (underscore). The chosen separator must also be concatenated: space || space.



3. Click OK
4. Add the Expression to the Field Page: Click Use as Field

On the Fields Page, edit and change the Field Column Heading text to EmployeeID/Record

5. Run and test the query.

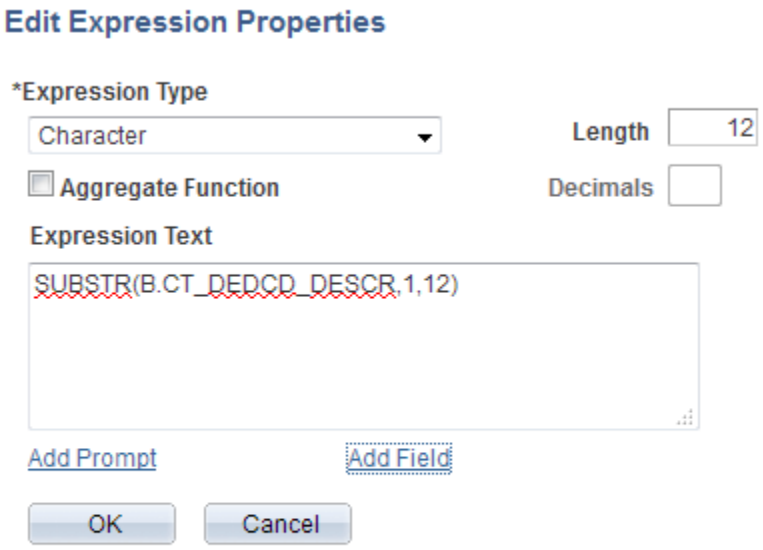
Substring

A Substring expression will only return characters of a field value you designate in the expressions syntax. Substring expressions are commonly used on fields where a portion of a field value is constant or definitive (i.e., DEPTID, BUSINESS_UNIT, ACCOUNT_CD, etc.)

Exercise Steps 2

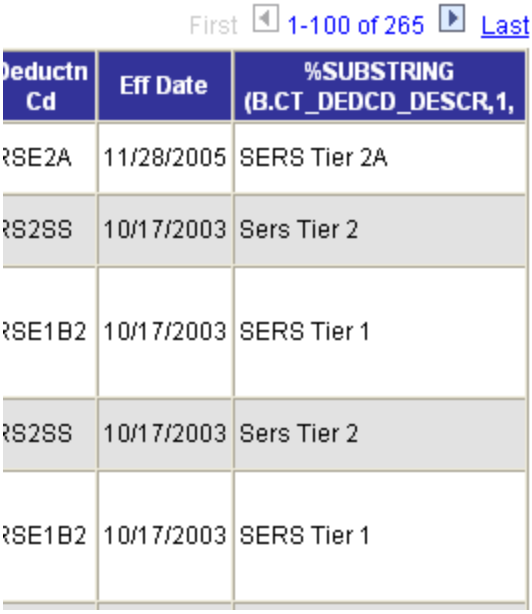
1. Locate query created in "Joining Tables, Simple Prompts" section above.
2. Add Expression:

Expression Type = Character; Length = 12; Expression Text = SUBSTR (B.CT_DEDCD_DESCR,1,12)



This substring expression will only return the first twelve characters of the deduction code description. The numerals 1,12 indicate the starting point and how many characters to display. Substring allows you to isolate part of a larger field (e.g., the first three characters of the DeptID field to designate agency).

3. Add the Expression to the Field Page: Click Use as Field
4. On the Fields Page, edit and change the Field Column Heading text to Retirement Plan
5. Run and test the query.



First 1-100 of 265 Last

Deductn Cd	Eff Date	%SUBSTRING (B.CT_DEDCD_DESCR,1,
RSE2A	11/28/2005	SERS Tier 2A
RS2SS	10/17/2003	Sers Tier 2
RSE1B2	10/17/2003	SERS Tier 1
RS2SS	10/17/2003	Sers Tier 2
RSE1B2	10/17/2003	SERS Tier 1

Decode

Decode allows you to create a field whose value is conditional upon a logical expression. The decode expression follows the if-then functionality of other software programs.

The general format is the following: DECODE (statement to evaluate, result to evaluate statement against, value if true, value if false).

The format for the Query tool is: DECODE(field name A, result 1,field name B, result 2). This reads If field name A =result1; return field name B; if not, then return result 2.

Exercise Steps 3

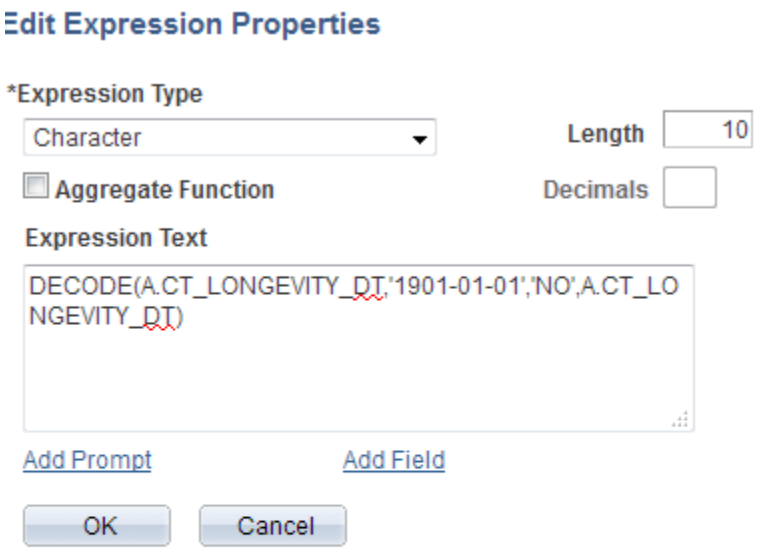
1. Locate query created in “Joining Tables, Simple Prompts” section above.

2. Add Expression: Expression Type = Character ; Expression Length = 10 ; Expression Text = DECODE(A.CT_LONGEVITY_DT, '1901-01-01', 'NO',A.CT_LONGEVITY_DT)

The expression field type and length are determined by the expected result.

Translates to: If the value for the field CT_LONGEVITY_DT is the bogus value 01-01-1901, then return NO, else return the value for CT_LONGEVITY_DT.

Note the single tics surrounding the ‘1901-01-01’and ‘NO’. These are text fields and must be designated as such. Also, note the format of the date field. The Oracle database converts dates to character fields.



3. Run and test the query.

Case

The case expression is similar to DECODE with the enhanced functionality of using condition types other than equal to.

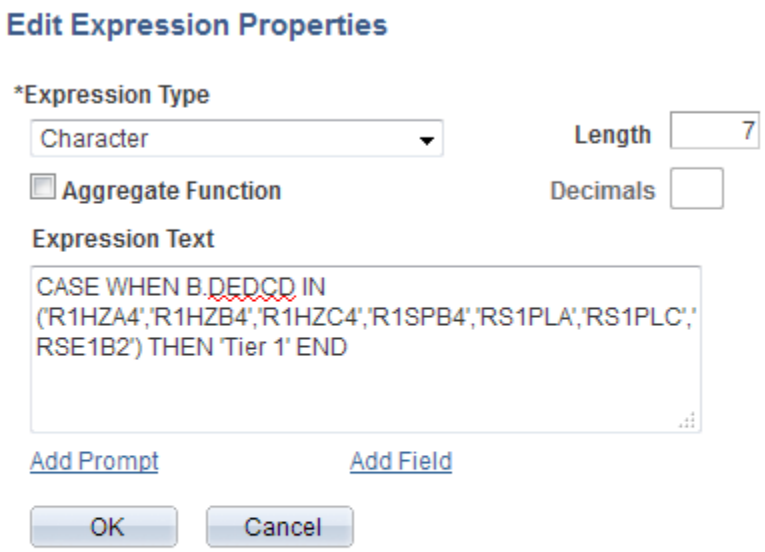
The general format is the following:
CASE WHEN statement to evaluate, condition type (eg. in, <>, >,<) result to evaluate statement against THEN value if true ELSE value if false END
The ELSE portion of the statement can be omitted. The return will be a blank field.

Fields created using the Case functionality can have aggregate functions applied to them.

Exercise Steps 4

1. Locate query created in “Joining Tables, Simple Prompts” section above.
2. Add Expression: Expression Type = Character ; Expression Length = 7 ; Expression Text =

CASE WHEN B.DEDCD IN
('R1HZA4','R1HZB4','R1HZC4','R1SPB4','RS1PLA','RS1PLC','RSE1B2') THEN 'Tier 1' END



Date Manipulation

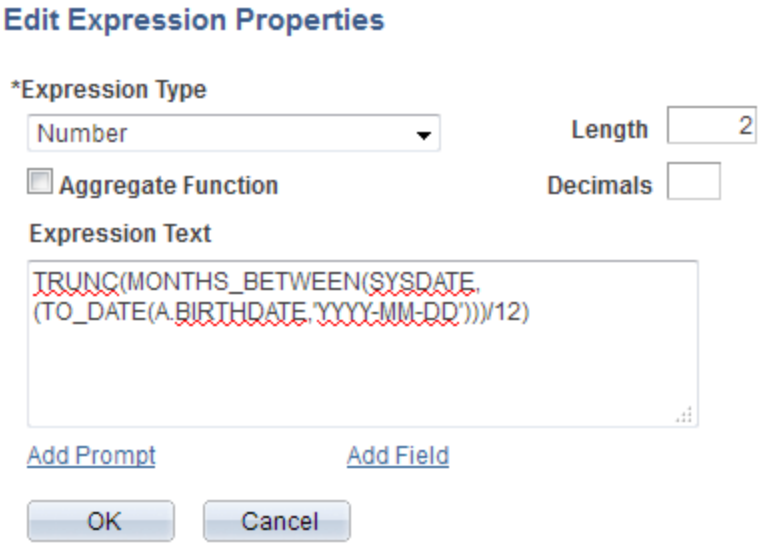
The primary obstacle to manipulation of dates through the query manager tool is the Oracle implicit conversion of dates to characters. To effectively work with dates, define the date format via the SQL tab using the TO_DATE command.

Exercise Steps 5

Scenario: This series of expressions will return an employee's age in years, months and days

1. Locate query created in "Joining Tables, Simple Prompts" section above.
2. Add an expression to return years.

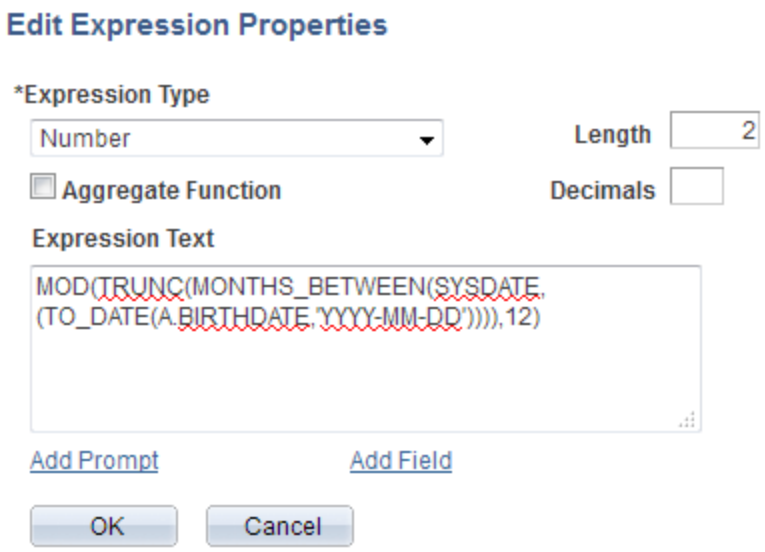
Expression Type = Number ; Expression Length = 2.0 ; Expression Text=
TRUNC(MONTHS_BETWEEN(SYSDATE,(TO_DATE(A.BIRTHDATE,'YYYY-MM-DD')))/12)



3. Click the use as field hyperlink
4. Rename column heading-Years
5. Run and test the query.
6. Add an expression to return months.

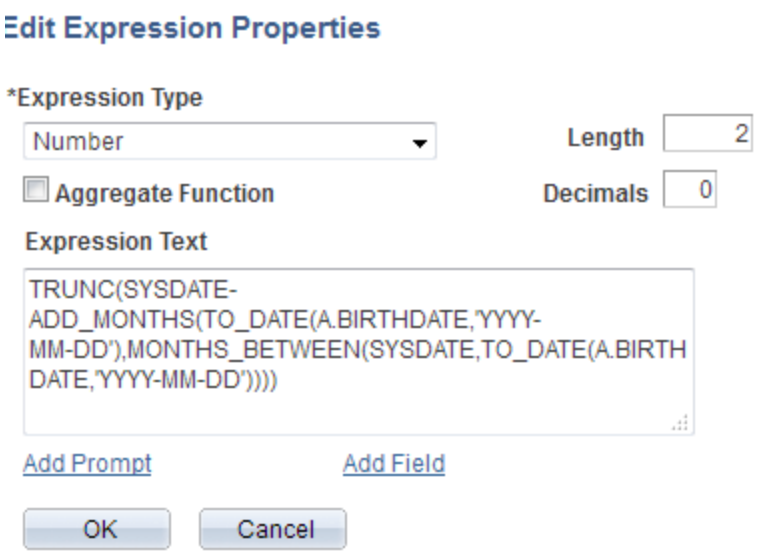
The command MOD returns the value of the remainder when dividing one number by another.

MOD(TRUNC(MONTHS_BETWEEN(SYSDATE,(TO_DATE(A.BIRTHDATE,'YYYY-MM-DD')))),12)

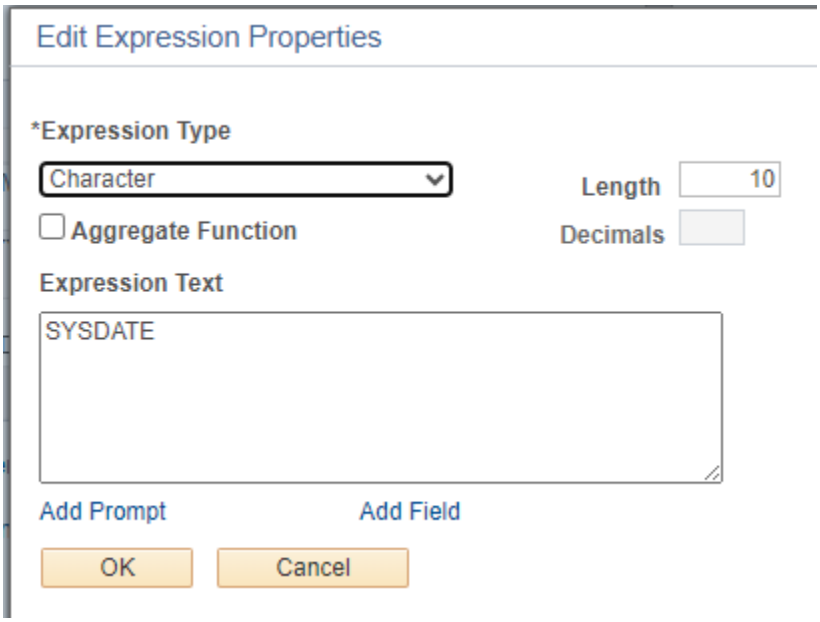


7. Click the use as field hyperlink
8. Rename column heading Months
9. Run and test the query.
10. Add an expression for Days

TRUNC(SYSDATE-ADD_MONTHS(TO_DATE(A.BIRTHDATE,'YYYY-MM-DD'),MONTHS_BETWEEN(SYSDATE,TO_DATE(A.BIRTHDATE,'YYYY-MM-DD'))))



11. Click the use as field hyperlink
12. Rename column heading Days
13. Run and test the query.
14. Add SYSDATE as a column.



15. Run and test the query.

Example public query using expressions: CT_CORE_HR_TL_SICK_REPORT

Note: When setting Criteria on a Date Time field, always use greater than or less than. Truncating the field and attempting to manipulate it as a simple Date field will be unsuccessful.

Subqueries

Subqueries Description

- A subquery is a query within a query.
- Subqueries enable you to compare the value for a field in the outer query to the results of the inner query.
- A subquery can retrieve only one data field from a single table. The subquery can contain a table join. However, criteria can be set up on many fields without using the results as a field.
- The results of the subquery are not displayed. The query results are limited by the results of the subquery.
- Single value subqueries use the condition types of exists or does not exist, in other words, the results are true or false.
- Using exists or does not exist requires a link between a field in the subquery and the outer query. You must set up table join criteria between the levels.

Exercise Steps

Scenario: Audit FICA exempt employees

Top Level

1. Add the table Required

A.**CTW_EMPLOYEE_VW** - Employee Information View

2. Add the fields

A.DEPTID
A.EMPLID
A.EMPL_RCD
A.NAME
A.CT_JOBCODE_DESCR
A.ORIG_HIRE_DT

A. FICA_STATUS_EE

3. Set up criteria

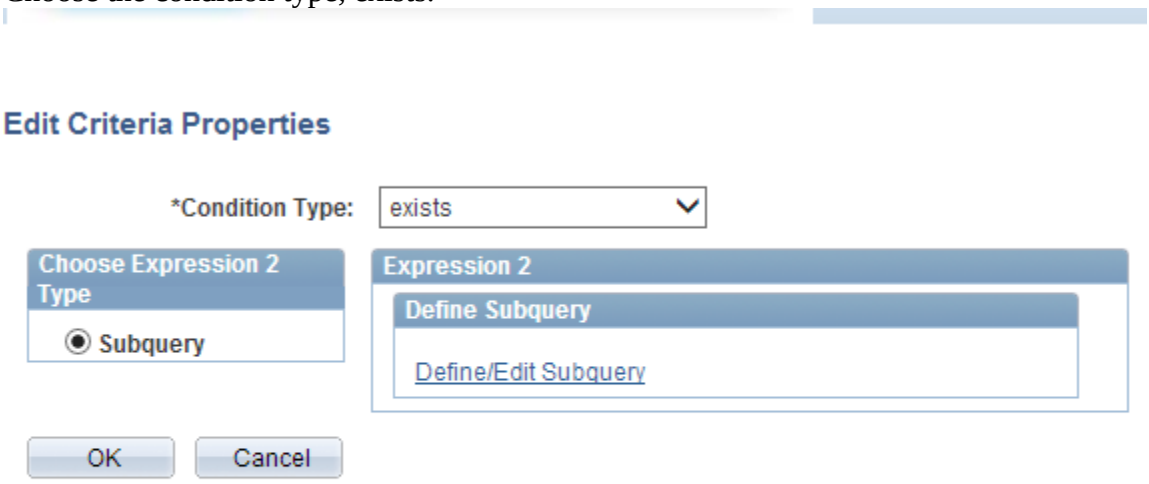
A.EMPL_STATUS in list (A,L,P,S)
A.FICA_STATUS_EE not equal to N
Prompt: A.DEPTID like :1

4. Edit the field FICA_STATUS_EE
   Choose Radio Button Long Translate value

Subquery

5. Set up the subquery, using the condition type exists

From the fields tab, click the add criteria icon for the field EMPLID.
Choose the condition type, exists.



6. Click the Define/Edit Subquery link to choose the table required for the subquery

7. Locate the record, **CTW_DTPY3PCT_VW** - Detail Payroll 3Pct View
******* This version of detailed pay has been chosen because it has no department security on it, so will neutralize the impact of the recent department consolidation.

8. Click add record

Note: The check off boxes to the left of the fields have been replaced with a Select hyperlink. Subqueries allow the selection of only one field.

9. Select EMPLID

10. Go to the Criteria tab and join the subquery record to the top level record.

Subquery tables must be joined to one of the outer query tables to retrieve accurate information

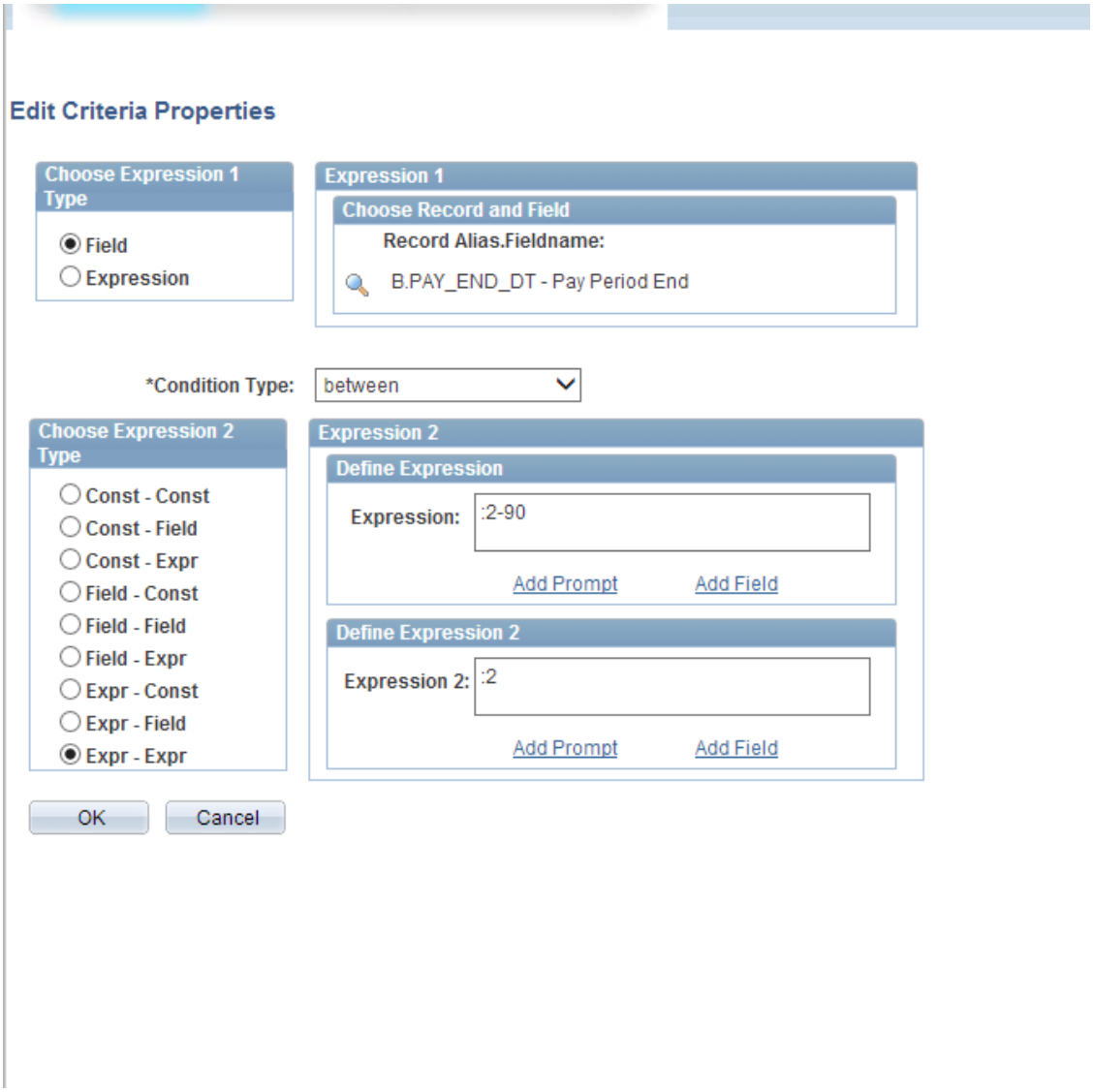
11. Set up Field joins

A.EMPLID=B.EMPLID
A.EMPL_RCD=B.EMPL_RCD

12. Set up all criteria on the subquery from the query tab, not displaying the results.

13. Create prompt on the field PAY_END_DT.

14. Establish between date range prompts. The logic is the employee is being paid sometime between Pay end date 90 days ago and the most recent (or the chosen pay end date).



15. Save and test the query.

Example public queries using subqueries:

CT_CORE_UNPROC_TIME
CT_CORE_UNPROC_TIME_TEST_NO_GR
CT_CORE_HR_PYRL_CHG_OTHR_AGY

Union Query

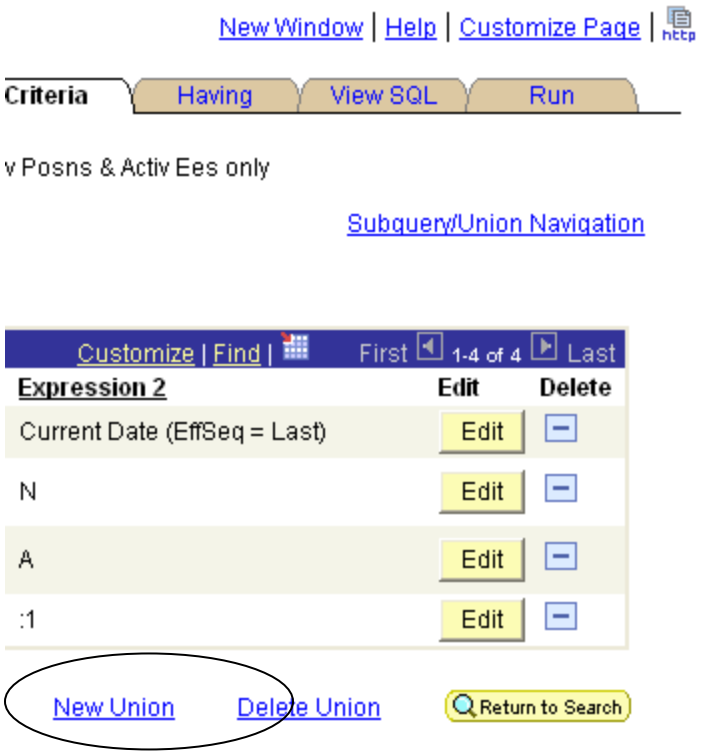
Union Query Details

- A union query is two select statements brought together in the same query, basically two queries running simultaneously.
- Use a union query to combine records that have no fields in common to retrieve similar values.
- Unions are especially valuable to avoid situations where a record join will produce inaccurate results. For example, when joining the employee and position tables, only those positions which are filled will be returned.
- If one simple rule is followed, union queries will always execute properly. The two portions of the query must have the same number of fields, in the same order like to like (field type and length). The field type must be exact and length similar. To achieve this, use literal expressions as placeholders.
- The "balancing" literals can be generic, character is ‘’(two single apostrophe marks), number is 0 (zero); or you can designate a word, between the two apostrophe marks (‘vacant’) or integer (9999) to clarify the report content.
- The table with the largest field sizes must be chosen as the top level of the query.
- Only the column names from the top level will display in the query output.
- Unions retrieve unique rows only. If both select statements retrieve the same row, the row will only appear once in the final output.
- You cannot use the long or short translate values in union queries.
- The limit is 12 unions, a total of 13 queries running at the same time.

Exercise Steps

Scenario: Build a query to identify all filled and unfilled positions.

1. Choose the table used for the top level
2. Add the required fields
3. Set up necessary criteria
4. Click hyperlink New Union



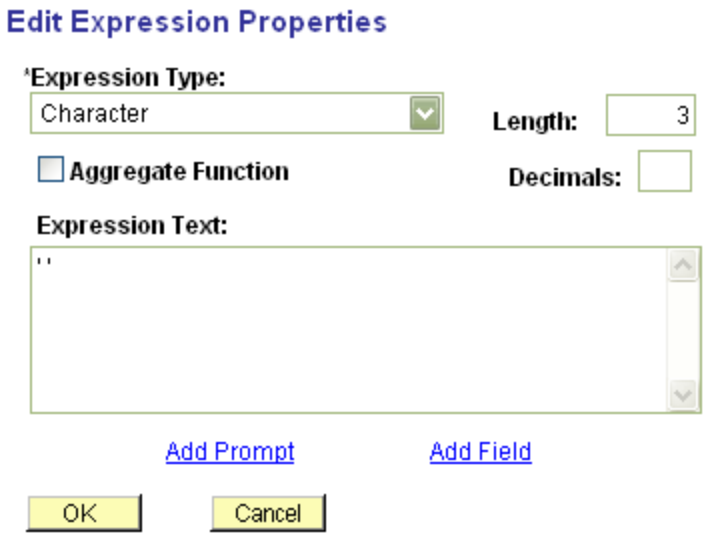
5. Choose the table for the Union 1 Level

6. Add the required fields.

If necessary, set up a literal expression to balance any fields unavailable on the table used in the Union 1 Level.

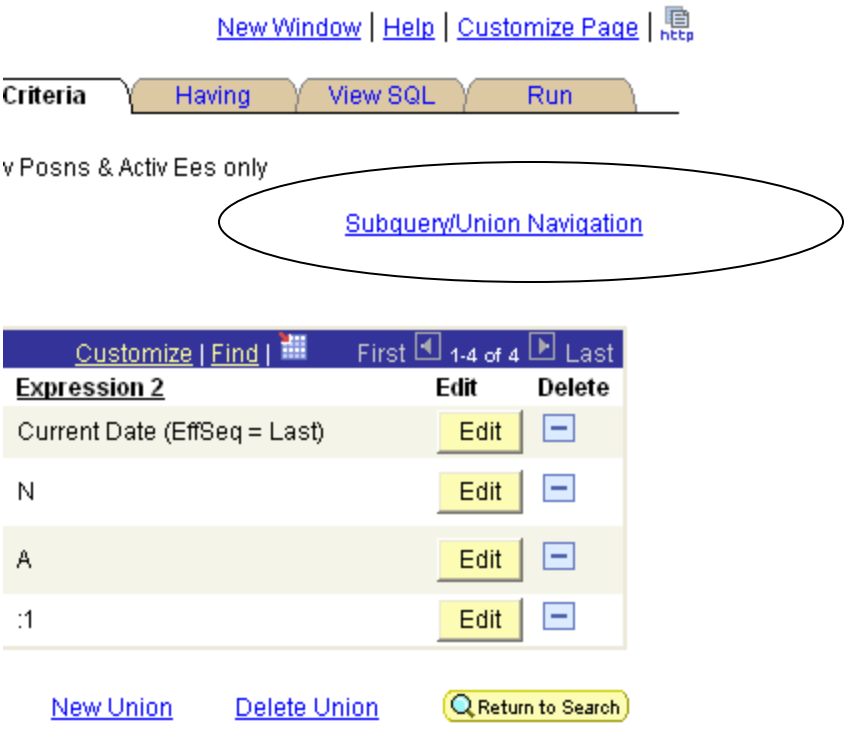
7. To set up a literal expression:

Add Expression:



The field format must match the format of the corresponding field in the Top Level of the query, both data type and length.

8. Add the Expression to the Field Page: Click Use as Field
9. Add criteria to the Union 1 Level.
10. Navigation between Top Level and Union 1 Level:



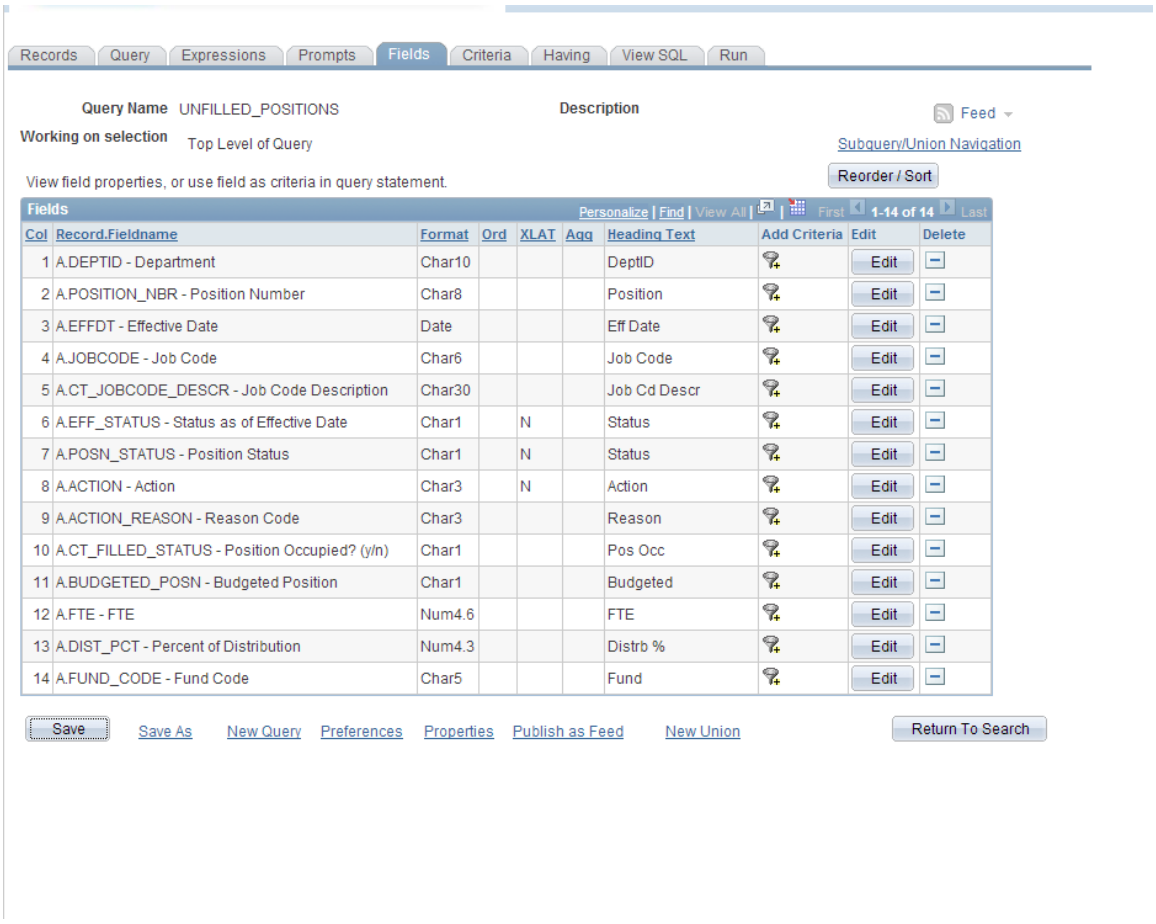
Top Level

Scenario: Produce a list of all active positions in an agency, both filled and unfilled.

1. Table Required:

A.CTW_POSITION

2. Fields Required:



3. Top Level Criteria

A.EFF_STATUS=A
A.POSN_STATUS=A

4. Prompts

A.DEPT like :1

Union 1 Level

1. Tables Required

C.CTW_POSITION
D.CTW_EMPLOYEE_VW

2. Fields Required

Choose fields to balance those in the top level.

***Add fields to identify the employee occupying the position and set up balancing literals in the Top level to indicate the position is vacant.

3. Set up literals if necessary

Union 1 Level Criteria

4. Field join

C.POSITION_NBR=D.POSITION_NBR

5. Criteria

D.EMPL_STATUS in list(A,L,P,S)

6. Prompt

C.DEPTID like :1

7. Save and test the query

Effective Date Manipulation

Scenario: Evaluate Effective Date Logic

1. Create a query using CTW_EMPLOYEE_VW

***Results**: Upon adding the record, the effective date pop up notification appears stating an effective date criteria has been automatically added for this effective dated record.*

2. Add the fields:
   EMPLID
   EMPL_RCD
   EFFDT
   EFFSEQ
   ACTION_DT
   ACTION
   ACTION_REASON
   EMPL_STATUS
   DEPTID

LOCATION
JOBCODE

3. Set criteria on the field DEPTID, hardcoded to your University/Community College/Agency.
4. Set Criteria on EMPL_STATUS; A,L,P,S.

Execute the query and note the row count.

5. Change the criteria for the EFFDT from
   EFFDT<=Current Date, EFFSEQ=Last
   To a constant value, 07-01-2023

   Execute and note the row count.

6. Return the EFFDT criteria to the default value,
   EFFDT<=Current Date, EFFSEQ=Last

7. Set Criteria on the ACTION field to a constant value, HIR

Execute and note the row count.

8. Delete the row of criteria, EFFDT<=Current Date, EFFSEQ=Last

9. Go to the Fields Tab, select the Edit Button and radio button MAX.

Execute and note the row count.

Special Attention needs to be paid to the relationship between date fields, especially with respect to agency consolidations.

When joining employee data tables with payroll tables, ALWAYS make the employee data table EFFDT<=PAY_END_DT or DUR.