

A Quick Guide to EPM Query Design and Analysis

Last Updated September 2015

OVERVIEW

Enterprise Performance Management (EPM) is a reporting portal that allows users to retrieve the data that stored in various reporting tables and to populate the results as designed. To ensure that the query design returns the values that are desired, users should carefully exercise the following principles:

- ❖ **Determining the purpose of the query.**
It is crucial to identify exactly what information is being sought and proper fields, parameters and logical conditions are used. A row of data will only return if it meets all the conditions, criteria and field requirements based on the query design.
- ❖ **Determining if production (real-time) data is needed.**
Data in HRMS, Financials, and Projects and Contracts tables are mostly refreshed on a nightly basis, Sunday thru Friday, and is at least one day old. Payroll data is only refreshed after the bi-weekly payroll cycle.

Refresh schedule and/or status are available online:

EPM bi-weekly	EPM Biweekly Table Refresh Schedule
HRMS	Current EPM HRMS Status by Table
Financials	Current EPM Financials Status by Table
Projects and Contracts	Current EPM Projects and Contracts Status by Table

- ❖ **Determining if a Public Query already exists.**
Users can access a listing and description of available public queries, categorized by subject areas, at the [Core-CT Catalog of Reports and Queries](#). To modify an existing Public Query, users must save the query as private.

If no public query exists, users can design a query using the logical progression of tabs in the Query Manager components, including the reporting table or tables, data fields, criteria, prompts, and expression. Additional elements of query design criteria include condition types, aggregate functions, and Having criteria.

Always test the query using known data. If a query returns no data or partial data, verify that data exists and then reevaluate the query design to ensure proper fields, parameters and logical conditions are used.

I. QUERY COMPONENT

Navigation: Main Menu > CORE-CT EPM > Reporting Tools > Query > Query Manager



A. Records Tab

From this page, users search and select the record/table(s) that contain the data fields (report columns) necessary to meet reporting needs by various categories. To find specific fields, navigate to the on-line EPM Data Dictionary: CORE-CT EPM > EPM Reporting Tools > Data Dictionary.

For a description of tables and their data fields, click one of the following links:

HRMS	HRMS Reporting Table Summary
Financials	Financials Reporting Table Summary
Projects and Contracts	Project Costing and Contracts Reporting Table Summary

ILLUSTRATION:

A report is needed to identify each active employee, by Department ID, who earns less than \$100,000 annually in Bargaining Units 01, 02 and 03 (Managerial Bargaining Units) for the calculation of 3% cost of living increase

- Users will first determine which reporting table or tables contain active employee data by Department and Bargaining Unit as well as salary information.
- The **HRMS** reporting table - **CTW_EMPLOYEE_VW** contains not only all active employees by Department and Bargaining Unit but also contains their annual rate of salary. Therefore, the **CTW_EMPLOYEE_VW** table is the only table needed to generate the Sample Report.

B. Query Tab

This page displays all the data fields associated with the table(s) chosen from the Records tab. Users select the fields by clicking the checkbox to the left of the field name. The final report will present the data associated with the fields selected for report columns.

Note: On this tab as well as the Criteria tab users can establish criteria without selecting the field to display.

ILLUSTRATION:

- The Sample Report will display the following fields: Employee ID (**EMPLID**), Name, Department (**DEPTID**), Employee Status (**EMPL_STATUS**), Union (**UNION_CD**), and Annual Salary (**ANNUAL_RT**).
- Listing of all current employees in a specific bargaining unit should have limiting criteria on union code; however, it may not be necessary to see the union code displayed for each employee.
- The following fields can be optional to provide more detail: Job Code (**JOBCODE**), Job Title (**CT_JOBCODE_DESCR**), and Location (**CT_LOCATION_DESCR**).

C. **Expressions Tab**

An expression is a user-defined field which can include many functions (i.e., calculations, 'if' and 'case' statements, field concatenations, sub strings, etc), similar to logical functions available in Excel.

ILLUSTRATION:

To establish a calculation function presenting the 3% cost of living increase for all active employees who earn less than \$100,000 annually in the Managerial Bargaining Units, users will create a field to multiply an employee's current annual rate by 3%.

- Click the Add Expression button on the Expressions Tab.
- Change the Expression Type to Signed Number with a length of 22 and decimal of 6.
- In the Expression Text box, enter the formula: ***A.ANNUAL_RT*1.03***.
- Click Ok; then click on the use as a field link.
- From the Fields tab, click the Edit button to change the Heading Text to 3% COLA.

D. **Prompts Tab**

A prompt provides the ability to allow the user to change criteria on a field at runtime. Prompts can either be established from this tab or by clicking the funnel icon located to right of the field name. Using prompts versus hard coding criteria is more efficient since a generic query can be developed and executed for a variety of parameter values.

Note: Prompts are displayed in the order in which they were created. Prompts cannot be reordered after creation. Additionally, multiple prompts cannot have the same text.

ILLUSTRATION:

The Sample Report's purpose is to return values based on Department. Since some agencies have multiple Department codes, a prompt can be established to allow the end user to determine from which Department code to generate the report. A simple prompt is easily established by clicking on the funnel icon right of the field, DEPTID, either on the Criteria Tab or on the Fields Tab.

- Click on the appropriate condition type.
- Choose either ***equal to*** (if the end user knows the Department code(s) exactly) or ***like*** (if the end users knows only part of the Department code(s)).
- Click on prompt in the Choose Expression Type 2 box.
- Click on the ***New Prompt*** link in the Expression 2, Define Prompt box.
- On the Edit Prompt Properties page, change the Heading Text to read ***Department is =*** (if condition type is set to equal to) or ***Department is like %*** (if condition type is set to like).
- Click OK twice.

E. **Fields Tab**

This page will display all the fields chosen from the Query tab, which will be the columns chosen for the final report. On this tab, the user can also –

- establish criteria by clicking the add criteria icon to the right of the field name;
- edit the field's properties as well as establish simple aggregate functions by clicking the edit button also located to the right of the field name;
- format the final report by clicking the edit/sort button in the upper right hand corner below the tabs to change the column order and/or sort the data in the final report.

ILLUSTRATION:

- First, review the fields chosen on the Query Tab for column order, sorting and readability. Since our Sample Report is based upon bargaining unit and annual salary, you may want to sort the data by Bargaining Unit then by salary and have both the current Annual Rate and 3% COLA Annual Rate side by side for comparison purposes. To change the order of the fields (the columns in your final report) or to sort the data, click on the Reorder/Sort button.
- Second, determine if simple criteria can be added to any field to limit the data to return only the necessary values.

F. **Criteria Tab**

Criteria should be utilized to limit data, where appropriate. Due to the large volume of data available in EPM as well as the size restrictions imposed by software programs such as Excel, criteria should be set to limit the query retrieves only the data that is required. Criteria can be established on this tab by clicking the add criteria icon/funnel located right of the field name.

ILLUSTRATION:

First, when searching for current employees by job class, criteria should be set to retrieve only active employees; otherwise, without the limiting criteria, the query would return all employees, regardless active or inactive, within a job class.

- Click on the funnel icon to the right of the field, ***EMPL_STATUS***.
- Change condition type to ***in list*** and click on the looking glass in the Expression 2, Edit List box. Click on the add value button for the following values: ***A-Active, L-Leave of Absence, P-Leave with Pay and S-Suspended***.
- Click OK twice.

Second, when selecting specific Managerial Bargaining Units, a list of values is not available.

- Click on the funnel icon to the right of the field, ***UNION_CD***.
- Change condition type to ***in list*** and click on the looking glass in the Expression 2, Edit List box.
- Manually enter each Bargaining Unit code separately. Enter ***01, 02*** and ***03*** for Managerial Bargaining Units.
- Click OK twice.

Lastly, to limit annual salaries to less than \$100,000.

- Click on the funnel icon right of the field, ***ANNUAL_RT***.
- Change the condition type to ***less than***.
- Enter ***\$100,000*** as a constant value.

Note: the condition types less than and greater than do not include the specified value.

G. **Having Tab**

A HAVING clause is similar to a Where clause for rows of data produced when aggregate functionality has been applied. In this case, the Sample Report's purpose does not indicate the need for Having criteria. However, if the Sample Report's purpose was slightly altered to determine the number of employees who earn more than \$70,000 but less than \$100,000 either currently or after the 3% COLA, Having criteria could be established to return this level of analysis.

H. View SQL Tab

The code produced by the Query Manger Tool to extract data, structured query language (SQL), is displayed on this tab. The SQL display for a query is generally used by the CORE-CT Help Desk in reviewing and diagnosing user query issues. The user cannot edit SQL through the Query Manager Tool.

I. Run Tab

Executes the query through the web and should only be used when testing very small data samples. The preferred methods of execution are clicking the Excel or HTML hyperlinks or by scheduling a query from the Query Scheduler link located right of the query name.

ILLUSTRATION: Sample Report

Records	Query	Expressions	Prompts	Fields	Criteria	Having	View SQL	Run		
DeptID = DAS%										
View All Rerun Query Download to Excel First 1-68 of 68 Last										
	DeptID	Location Descr	ID	Name	Job Code	Job Cd Descr	Status	Union Code	Annual Rt	3% COLA
2	DAS23000				0143MP	EqualEmployOppSpec2	A	02	68270.031000	70318.131930
3	DAS23000				6597MP	CommsSpec(Mgr)RC	A	02	78763.014000	81125.904420
4	DAS23000				6597MP	CommsSpec(Mgr)RC	A	02	85436.001000	87999.081030
12	DAS23000				7725MP	SeniorPolicyAdvisor	A	02	99559.233000	102546.009990
13	DAS23000				7539CA	Secretary2	A	03	45747.036000	47119.447080
14	DAS23000				7086CA	OfficeAssistant	A	03	49108.194000	50581.439820
15	DAS23000				6161CR	HumanResourcesAssistant	A	03	52761.150000	54343.984500
16	DAS23000				7539CA	Secretary2	L	03	52996.050000	54585.931500
17	DAS23000				6163MP	HumanResourcesAssociate	A	03	53168.000000	54763.040000
18	DAS23000				5989VR	LeadershipAssociate(Conf)	A	03	54657.054000	56296.765620
19	DAS23000				6160CR	PersonnelAideRC	A	03	55129.203000	56783.079090
20	DAS23000				7539CA	Secretary2	A	03	55910.115000	57587.418450
21	DAS23000				7539CA	Secretary2	A	03	55910.115000	57587.418450

II. ADDITIONAL CRITERIA

J. Condition Types

Not all condition types are applicable to every data field. For example, condition types “*is null*” on “*not null*” only refers to date field types. Wildcards, “%” and “_” (underscore) are only used with the condition types *like* and *not like*.

A listing and description of [Queries: Condition Type](#) is available online.

K. Effective Date

Some tables have effective date logic built into the reporting table design. When data is updated by adding a row, there is an effective date associated with each row; the system retains the previous row as history. Effective date logic allows for the return of the most current information and also can provide historical or future dated rows. To retrieve the most current information, effective dated logic asks for the most current row dated less than or equal to today.

L. Aggregate Functions

Query Manager can perform simple mathematical functions. Users may apply simple aggregate functions to appropriate fields when necessary. Examples of aggregate functions are as follow:

Sum	Adds the numerical values from each row and displays the total
Count	Counts the number of rows
Min	Compares the values from each row and displays the lowest value
Max	Compares the values from each row and displays the highest value
Average	Adds the numerical values from each row and divides the total by the number of rows

III. **ADVANCED QUERY DESIGN**

Complicated queries may need to utilize sub-queries and unions in addition to table joins. Please refer to [Advanced EPM Reporting Class Notes](#) for helpful tips.

Joining Tables

Careful attention must be given when joining two or more tables together to extract all the data needed. Not all tables can be joined. To ensure a join is possible as well the correct field joins between tables, please refer to [Reporting table Indexes and Join Criteria](#) before joining tables.

Sub-Query and Union

A **Subquery** is used by the outer query as a comparison value for a selection criterion. A subquery contains table joins and criteria that can be set up on many fields but the results are not displayed. A table join criteria is required to link between a field in the subquery and the outer query. Consequently, the output of the outer query may be limited by the results of a subquery. Condition types of a subquery include:

- ❖ “*exists*” or “*does not exist*” for single value subquery.
- ❖ “*in list*” or “*not in list*” for multiple value subquery.

Note: Change the condition type, if getting the error message “not a single value subquery” when setting up a “exists”/“ does not exists”.

A **Union** query is created when the data from multiple tables is required but cannot be successfully joined. A union query can combine records that have no fields in common to retrieve similar values and execute simultaneously in the top level of the query. It is crucial to have an equal number and matching field types and length similarity of fields in both the top level of the query and in the union level. Prompts and literal expressions, functioning as placeholders, are required when creating a union query.

Note: literal expression can be generic, i.e., character is ‘ (two single apostrophe marks), and number is 0 (zero).

IV. SCHEDULING QUERY

If there is a large amount data to be returned and or if there is a recurrent need to execute the report, queries should be run in process scheduler to maximize efficiency and to achieve optimum query run-times. Reports generated from a scheduled query can be shared with any CORE CT user via the query distribution functionality. Helpful tips are available on [Scheduling Queries](#).

V. ADDITIONAL RESOURCES

[EPM Ad Hoc Reporting Job Aids
Tips and Tricks](#)