



TRAINING GUIDE

Filters and Queries

Desktop and Dashboard



Filters and Queries: Desktop and Dashboard

In this workbook we will discuss the intricacies of creating advanced filters and queries. We will demonstrate the different types of queries and how to retrieve the data that you need. We will also discuss creating filters for the dashboard and how this differs with creating filters in the desktop. Examples will be provided to aid you in creating your own filters in the dashboard and the desktop.

Table of Contents

Advanced Filters in the Desktop	2
Defining Relationships.....	2
Properly Formed Queries	2
Creating Filters using “or”	3
String Values	3
Boolean Fields	3
Date Fields	3
Querying for Null and Non-Null Values	4
Wildcards in Filters	4
Advanced Filters and Crystal Reports	5
Conversion Convention Examples	6
Special Keywords.....	6
Examples:.....	8
Additional Information	9
Special Filters for the Dashboard.....	10
Acquiring Filters.....	10
Restrictions.....	11
Advanced Filters: Web vs. Desktop	11
Web Application Filter Examples:	11
Desktop Application Filter Examples:	11
Default Joins	12
Filtering on Child Records	12
More Filter Practice	13

Advanced Filters in the Desktop

Defining Relationships

Sometimes it is necessary to query information in a table other than the main record's table. In that case, you need to define a relationship between the main record's table and the "child" table. The relationship will link the two tables together based on the information that is contained in both tables. When you define a relationship, it is recommended that you use a "WHERE" clause, which will retrieve records that fulfill a specific criteria.

Here are a few examples of this query method from the Work Order Module.

1. This query will return all of the work order records with Request Comments Keyword Lookup:
 - `WKORDER WHERE WO_ID IN (SELECT CO_REC_ID FROM WKMEMO WHERE CO_TABLE = 'WKORDER' AND CO_FIELD = 'WO_MEMO1' AND CO_TEXT LIKE ('%KEYWORD%'))`
2. This query will result in all of the work orders where there are work order locations with no building number defined and no X and Y Coordinates defined and a status of complete:
 - `WKORDER WHERE WO_ID IN (SELECT WL_WO_ID FROM WKORDERLOC WHERE WL_ADR_BDG IS NULL and WL_X_COORD IS NULL and WL_Y_COORD IS NULL) AND WO_STAT_CD>949`

Properly Formed Queries

Queries must also be well-formed, and any queries containing OR statements must be properly encapsulated in parenthesis.

Example #1: All work orders that have a Cause Code of "1" and any additional work orders that were still open and have either a Category code of "6000" or a Problem Code of "200".

INCORRECT

- `WKORDER WHERE (WO_STAT_CD<950 AND WO_CAT_CD = '6000' OR WO_PROB_CD = '200') OR WO_CAUS_CD = '1' ORDER BY WO_STRT_DT`

CORRECT

- `WKORDER WHERE ((WO_STAT_CD<950 AND (WO_CAT_CD = '6000' OR WO_PROB_CD = '200')) OR WO_CAUS_CD = '1') ORDER BY WO_STRT_DT`

Example #2: All work orders that are still open OR have a value in the User 6 field other than "0". And in either case, have a Problem Code of "200".

INCORRECT

- `WKORDER WHERE WO_STAT_CD <950 OR WO_USER6 <> 0 AND WO_PROD_CD = 200`

CORRECT

- `WKORDER WHERE (WO_STAT_CD<950 OR WO_USER6<>0) AND WO_PROD_CD = 200`

Creating Filters using "or"

"OR" can be used in a statement when you are looking for records where any one of several conditions is met.

- This filter will give you all of the records that either have a status of complete or a status date less than 8/1/2010.

```
WKORDER WHERE WKORDER.WO_STAT_TY = 'Complete' OR WKORDER.WO_STAT_DT < '08/01/2010'
```

- This filter will give you all of the records that fit any one of these situations: the Work Order number ends in 0001, the status of the work order is complete, or the category type is not Backflow preventers or Call Center.

```
WKORDER WHERE WKORDER.WO_NUMBER LIKE '%-0001' OR WKORDER.WO_STAT_TY='Complete' OR  
WKORDER.WO_CAT_TY NOT IN ('Backflow Preventers', 'call Center')
```

String Values

String values must be enclosed in single quotes: 'MyString'.

Boolean Fields

When querying for Boolean fields, use "<> 0" to retrieve TRUE values and "= 0" to return FALSE values.

Date Fields

The use of date fields must conform to the proper format for the database platform you use. The following table outlines date formats for each database platform.

Note: Rarely (if ever) would a filtered on field equal the current Date AND Time; thus, the following Date functions are designed to return the current Date only.

Database Platform	Format
Microsoft Access:	
Date Format:	#1/1/2001#
Current Date Function:	[FIELD] = Date()
Microsoft SQL Server	
Date Format:	'1/1/2001'
Current Date Function:	[FIELD] = CONVERT(VARCHAR(10), GETDATE(), 101)
Oracle	
Date Format:	'1/1/2001'
Current Date Function:	[FIELD] = TRUNC(SYSDATE)

Note: Any date with single quotes around only works for SQL or Oracle syntax.

Querying for Null and Non-Null Values

To Query for NULL values use the syntax [FIELD] IS NULL or its complement [FIELD] IS NOT NULL.

- For SQL Server and Microsoft Access, if the field is a character or text field, you may also want to include the following syntax [FIELD] = " or its complement [FIELD] <> ". Proper syntax must be used to formulate this OR statement. For example, ([FIELD] IS NULL OR [FIELD] = ")
- Oracle does not support the following queries [FIELD] = " , or [FIELD] <> ". For Oracle use the NULL portion of the query only.

Wildcards in Filters

For filters that require wildcards for a single character or many character positions, use the following:

SQL Server and Oracle Wildcards

- Single Position - Underscore (_)
- Many Consecutive Positions - Percent Symbol (%)

Access Wildcards

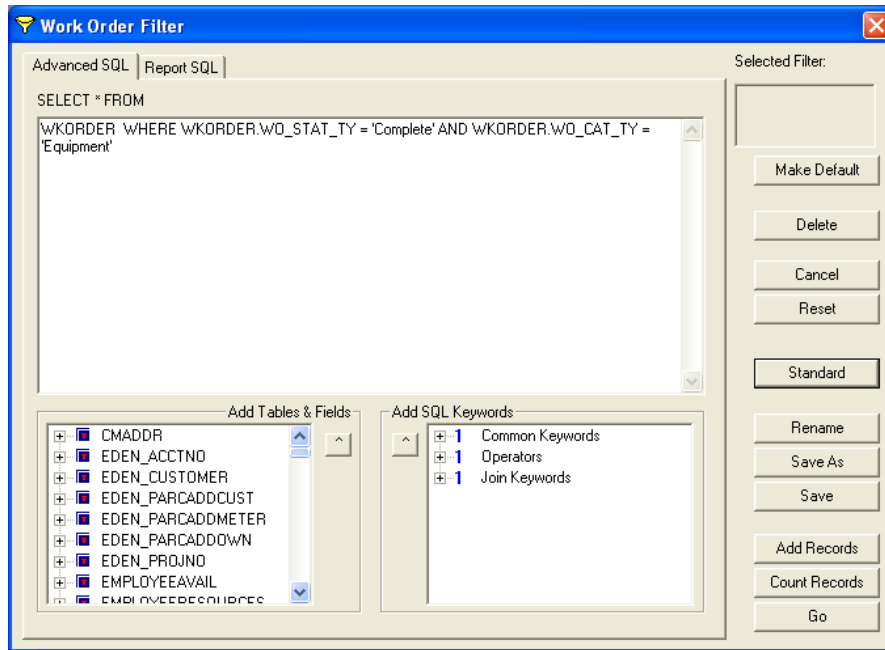
- Single Position - Question Mark (?)
- Many Consecutive Positions - Asterisk (*)

Notes: _____

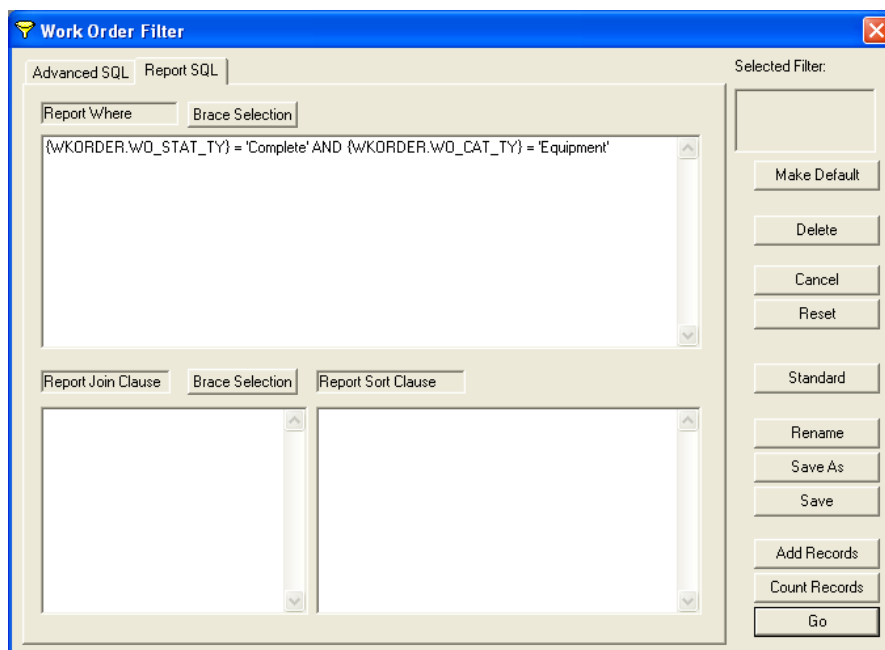
Advanced Filters and Crystal Reports

While standard filters are automatically converted into the correct syntax for Crystal Reports, advanced filters must be manually converted using the filter window within *Lucity*.

Here is an example of an advanced filter for a status of complete and the category is Equipment:



Here is the same filter, but in Crystal syntax:



Notice that the `TABLENAME.FIELDNAME` is within French brackets in crystal syntax. When you create an advanced filter, you can click on the Report SQL tab to convert the statement to crystal syntax. Below are a few examples of the conversion convention differences between Lucity convention and Crystal Reports convention.

Conversion Convention Examples

Lucity Convention	Crystal Reports Convention
TABLERNAME.FIELDNAME	{TABLERNAME.FIELDNAME}
In List ('Item 1', 'Item 2', 'Item 3')	{TABLERNAME.FIELDNAME} In ['Item 1', 'Item 2', 'Item 3']
Not In List ('Item 4', 'Item 5')	Not ({TABLERNAME.FIELDNAME} In ['Item 4', 'Item 5'])
> = #9/25/2010# (for Access)	> = Date(2010,9,25)
< '9/25/2010' (for SQL/Oracle)	< Date(2010,9,25)
> #2:00:00 PM# (for Access)	> Time(2:00:00PM) or > Time(14,0,0)

Special Keywords

The following keywords can be manually entered into the advanced filter.

Special Keyword	Description
'%FISCALYEAR%'	You must own the <i>Work</i> module to use this string. It uses the current fiscal year from the <i>Planning and Budgeting, Fiscal year</i> module.
'%CURRENTYEAR%'	Inserts the current calendar year.
'%CURRENTQUARTER%'	Inserts the current quarter.
'%CURRENTDAY%'	Inserts the current day.
'%CURRENTWEEK%'	Inserts the current week.
'%FIRSTHALFOFTHEMONTH%'	Inserts the first half of the month (days 1-15).
'%SECONDHALFOFTHEMONTH%'	Inserts the second half of the month (days 16-31).
'%CURRENTMONTH%'	Inserts the current month.
'%ONEWEEKSAHEAD%'	Inserts the next week.
'%TWOWEEKSAHEAD%'	Inserts the next two weeks.
'%ONEWEEKADEADANDBEFORE%'	Inserts the next week plus previous timespan. Ex. You may use this string to query for all existing open work orders as well as those that are open/due in the next week.
'%TWOWEEKSAHEADANDBEFORE%'	Inserts the next two weeks plus previous timespan.
'%LOGGEDINUSER%'	Allows you to filter on the logged in user.
'%PREVIOUSYEAR%'	Inserts the previous year

'%PREVIOUSMONTH%'	Inserts the previous month
'%PREVIOUSWEEK%'	Inserts the previous week
'%PREVIOUSDAY%'	Inserts the previous day.

Note: When using the reserved words above, make sure that you enter them in all uppercase letters and enclose them in percent symbols (%) and single quotes ('). Also, these reserved words can be used in the desktop app for advanced filters. Use an = sign with all the reserved words, "WO_STAT_DT = '%PREVIOUSDAY%'". However, you can use the greater than or less than signs when using these three keywords: CURRENTYEAR, CURRENTMONTH, or CURRENTDAY.

Note: These keywords are only supported for SQL and Oracle.

Work Order Filter

Advanced SQL | Report SQL

SELECT * FROM
WKORDER WHERE WKORDER.WO_MOD_DT = '%CURRENTDAY%' AND WO_CAT_TY LIKE '%ADM%'

Add Tables & Fields

- WKWORSTK
- WKWOTRAK
- WKWOTSK
- WKWTLOSS
- WKWTLOSSWO
- WTCMDEV
- WTMETDEV
- WTRMDEV

Add SQL Keywords

- Common Keywords
- Operators
- Join Keywords

Selected Filter: NEW

Make Default
Delete
Cancel
Reset
Standard
Rename
Save As
Save
Count Records
Go

Notes: _____

Examples:

1. Queries that filter based on a child record

- EFEEQUIP WHERE EFEEQUIP.GE_ID IN (SELECT EFREAD.ER_GE_ID FROM EFREAD)
- This will return equipment records that have any “reading” records associated with them.

2. Sign Regulations Key Work Search in Reg Location Comment Box

- STSIGNREG WHERE SR_ID IN (SELECT CO_REC_ID FROM STMEMO WHERE CO_TABLE = 'STSIGNREG' AND CO_FIELD = 'SR_MEMO2' AND CO_TEXT LIKE ('%40TH ST%'))

(SQL Server Only)

3. Water Hydrant flow Tests: Water Hydrants with or without Hydrant Test in a Date Range

- WTHYDRNT WHERE HY_ID IN (SELECT HT_HY_ID FROM WTHTEST WHERE HT_TEST_DT > '7/1/2001' AND HT_TEST_DT < '9/1/2009')
- WTHYDRNT WHERE HY_ID NOT IN (SELECT HT_HY_ID FROM WTHTEST WHERE HT_TEST_DT > '7/1/2001' AND HT_TEST_DT < '9/1/2009')

4. Work orders with Request Comments Keyword Lookup

- WKORDER WHERE WO_ID IN (SELECT CO_REC_ID FROM WKMEMO WHERE CO_TABLE = 'WKORDER' AND CO_FIELD = 'WO_MEMO1' AND CO_TEXT LIKE '%KEYWORD%')
- WKORDER WHERE WO_ID IN (SELECT CO_REC_ID FROM WKMEMO WHERE CO_TABLE = 'WKORDER' AND CO_FIELD = 'WO_MEMO1' AND CO_TEXT LIKE ('%LIGHT%'))

5. Find open work orders started within 7 days ago

- WKORDER WHERE WKORDER.WO_STAT_CD < 950 AND WKORDER.WO_STRT_DT < DATEADD("D",-7,GETDATE())

OR

- WKORDER WHERE WKORDER.WO_STAT_CD < 950 AND WKORDER.WO_STRT_DT < DATEADD(D,-7,GETDATE())

(For SQL Server or Oracle)

- WKORDER WHERE WKORDER.WO_STAT_CD < 950 AND WKORDER.WO_STRT_DT < DATEADD('d',-28,NOW)

(For Access)

6. Employee Certifications Overdue

- WKEMPCRT WHERE (WKEMPCRT. EC_ID) IN (SELECT ER_EC_ID FROM WKEMPCRTE WHERE (WKEMPCRTE.ER_EXP_DT > DATE()))

(For Access)

- WKEMPCRT WHERE (WKEMPCRT. EC_ID) IN (SELECT ER_EC_ID FROM WKEMPCRTE WHERE (WKEMPCRTE.ER_EXP_DT > current_timestamp))

(For SQL Server and Oracle)

7. Fleet on Work order with Certain Status

- EFFLEET WHERE (EFFLEET. FL_ID) IN (SELECT AS_INV_ID FROM WKWOASSET WHERE (WKWOASSET. AS_WO_ID IN (SELECT WO_ID FROM WKORDER WHERE (WKORDER. WO_STAT_CD > 1 AND WKORDER. WO_STAT_CD < 949) AND (WKORDER. WO_INV_ID=32))))

8. Parts with average cost more than 10 percent off what a recalc avg. would be

- PTPARTS, PTLOCATION WHERE PA_DEF_LC=PL_ID AND PA_ID IN (SELECT PA_ID FROM PTPARTS INNER JOIN PTPARTCOST ON PA_ID = PC_PA_ID INNER JOIN (SELECT PC_PA_ID, CASE WHEN SUM(PC_CURRQTY) = 0 THEN 0 ELSE SUM(PC_CURRQTY*PC_COSTUNT)/SUM(PC_CURRQTY) END AS AVGCOST FROM PTPARTCOST GROUP BY PC_PA_ID) T2 ON PTPARTCOST. PC_PA_ID = T2. PC_PA_ID WHERE ((PTPARTCOST. PC_COSTAVG > (AVGCOST*1.1)) OR (PTPARTCOST. PC_COSTAVG < (AVGCOST*.9))) AND AVGCOST <> 0)

(For SQL Server or Oracle)

This was needed to find parts where the calculated average cost was more than 10% off of what the average cost would be if recalculated based on on-hand inventory. This was due to the inability of the program to 'force' recalculating average cost. If somebody had a data entry error with part cost, the average cost would be irrevocably skewed.

Additional Information

Many books and web sites are available for SQL query syntax. Keep in mind that neither SQL Server, Access, nor Oracle fully conforms to SQL ISO standard, but information about proper syntax for individual platforms is also available on the web and in books.

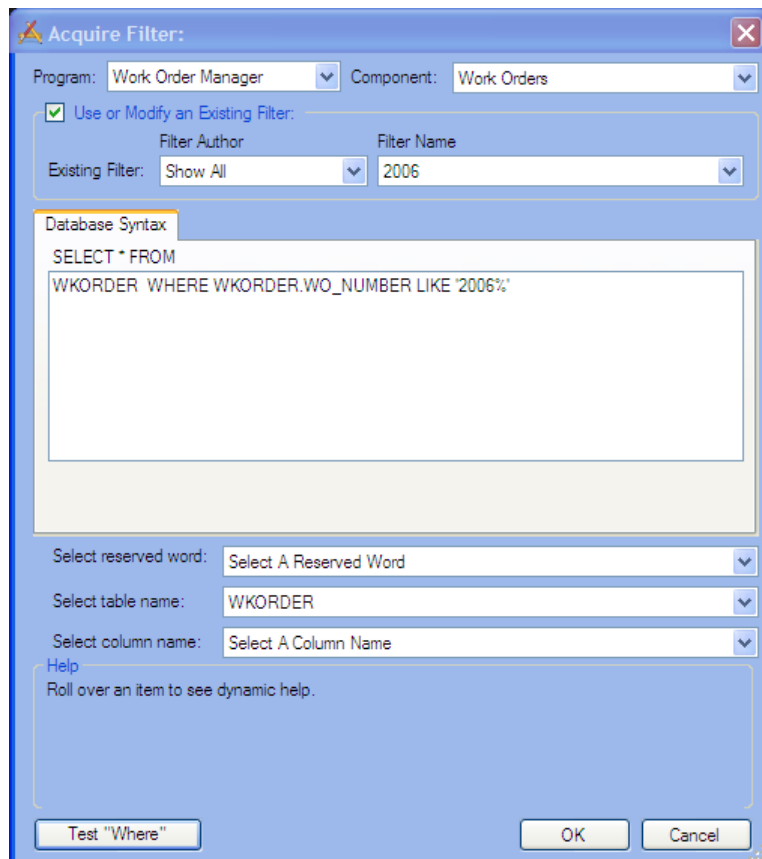
Notes: _____

Special Filters for the Dashboard

Acquiring Filters

To add a filter to a plugin, complete the following steps:

Click the *Acquire* button. The following window will appear:



1. Select a program and module from the drop down lists.
2. You have two options when choosing a filter:
 - You can type a filter directly in the large text field.
 - You can select a pre-defined filter from the desktop application by marking the *Use or Modify an Existing Filter* checkbox and then choosing a filter author and filter name from the drop down lists.

Note: Most users will select their own login ID as the filter author. This Filter Name drop-down will then display only the filters created using that login ID.

3. In the example above we have selected a pre-defined filter. The SQL statement for this filter is populated in the large text field on the Database Syntax tab. You'll use this tab to edit the filter's SQL statement if needed. By editing the text, it will change what is shown on the Dashboard page; it does not impact or change the original filter inside the desktop application.

4. You can test the filter against the desktop application data by clicking on the Test "Where" button. This will show how many records are returned when the filter is processed and indicate if any errors are detected.
5. Once you are satisfied with your filter, press OK.

Restrictions

Dashboard filters cannot contain ORDER BY clauses. The program will strip out the ORDER BY clause if you try to use one.

Advanced Filters: Web vs. Desktop

We've provided several sample Advanced Filters below. Note that the difference between advanced filters used in the Desktop application and Web application is in the syntax used for joining tables.

Web Application Filter Examples:

Select all Work Orders with a TASK CODE = 'EVNINT01'

- WKORDER WHERE WKORDER.WO_ID IN (SELECT WKWOTSK.WT_WO_ID FROM WKWOTSK WHERE WT_TASK_CD='EVNINT01')

Select all Work Orders with an EMPLOYEE resource named 'CHARLIE BROWN'

- WKORDER WHERE WKORDER.WO_ID IN (SELECT WKRESRCE.WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD=1 AND WR_RSRC_TY='CHARLIE BROWN')

Select all Work Orders with an EQUIPMENT resource named '2006 DODGE RAM'

- WKORDER WHERE WKORDER.WO_ID IN (SELECT WKRESRCE.WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD=2 AND WR_RSRC_TY='2006 DODGE RAM')

Desktop Application Filter Examples:

Select all Work Orders with a TASK CODE = 'EVNINT01'

- SELECT * FROM WKORDER WHERE WO_ID IN (SELECT WT_WO_ID FROM WKWOTSK WHERE WT_TASK_CD = ' EVNINT01')

Select all Work Orders with an Employee resource named 'CHARLIE BROWN'

- SELECT * FROM WKORDER WHERE WO_ID IN (SELECT WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD = 1 AND WR_RSRC_TY = 'Charlie Brown')

Select all Work Orders with an EQUIPMENT resource named '2006 DODGE RAM'

- SELECT * FROM WKORDER WHERE WO_ID IN (SELECT WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD = 2 AND WR_RSRC_TY = '2006 Dodge Ram')

Default Joins

The Lucity Administration program and the Filters in the web application will provide the default joins for every module where it is required.

For example:

- The default join shown for Fleet Fueling will be:
 - `EFFUEL LEFT JOIN EFFLEET ON FU_FL_ID = FL_ID`
- The default join shown for Equipment Warranty will be:
 - `EFWARR LEFT JOIN EFEQUIP ON WA_ASSET_ID = GE_ID WHERE WA_TYPE_CD = 9`

Note: These are examples of child modules that need to show data from their parent modules.

Filtering on Child Records

To filter on child data while looking at data in a parent, you should always use the "IN" syntax for joins:

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKWOTSK.WT_WO_ID FROM WKWOTSK WHERE WT_TASK_CD='EVNINT01')`

However, if you want to be able to drill on columns from a child table in a "data drill", then you can use the following syntax:

- `WKORDER INNER JOIN WKWOTSK ON WKORDER.WO_ID = WKWOTSK.WT_WO_ID WHERE WT_TASK_CD='EVNINT01'`

Note: When you use an inner join filter and open the records in the desktop, there will be one record for each task rather than one record for each work order. (I.e. if one work order has three tasks with the EVNINT01 task code; there would be 3 different work orders to navigate through in the desktop).

Notes: _____

More Filter Practice

Example # 1

Select all Work Orders with a MATERIAL resource named 'Aerosol Paint can'

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKRESRCE.WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD=3 AND WR_RSRC_TY='Aerosol Paint can')`

Example # 2

Select all Work Orders with a FLUID resource named '10W-30'

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKRESRCE.WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD=4 AND WR_RSRC_TY='10W-30')`

Example # 3

Select all Work Orders with a CONTRACTOR resource named 'TODDS TOWING'

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKRESRCE.WR_WO_ID FROM WKRESRCE WHERE WR_RTYP_CD=5 AND WR_RSRC_TY='TODDS TOWING')`

Example # 4

Select all Work Orders with a 'WATER HYDRANT' Asset that has a Common ID of 5261.

Note: the AS_CAT_INV of 12 represents 'Water Hydrants'. These numbers can be found in WKCATINV.

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKWOASSET.AS_WO_ID FROM WKWOASSET WHERE AS_CAT_INV='12' AND AS_LINK1='5261')`

Example # 5

Select all Work Orders with a 'SEWER PIPE' Asset that has an Upstream ID of TEST1 and a Downstream ID of TEST2.

Note: the AS_CAT_INV of 6 represents 'Sewer Pipe'. These numbers can be found in WKCATINV

- `WKORDER WHERE WKORDER.WO_ID IN (SELECT WKWOASSET.AS_WO_ID FROM WKWOASSET WHERE AS_CAT_INV='6' AND AS_LINK1='TEST1' AND AS_LINK2 = 'TEST2')`

Example # 6

Employee Certifications Overdue (SQL Server)

(Work>Employee/Cert Classes>Certifications)

- `WKEMPCRT WHERE (WKEMPCRT.EC_ID) IN (SELECT ER_EC_ID FROM WKEMPCRTE WHERE (WKEMPCRTE.ER_EXP_DT < current_timestamp))`

Example # 7

Fleet out of service (by being on a Work Order with a status of either 104-Fleet On-Going, 105-Fleet Waiting On Parts, or 106-Fleet Waiting On Vendor) (SQL Server)

(Equipment>Fleet>Fleet)

- `EFFLEET WHERE (EFFLEET.FL_ID) IN (SELECT AS_INV_ID FROM WKWOASSET WHERE (WKWOASSET.AS_WO_ID IN (SELECT WO_ID FROM WKORDER WHERE (WKORDER.WO_STAT_CD > 103 AND WKORDER.WO_STAT_CD < 107) AND (WKORDER.WO_INV_ID=32))))`

Example # 8

All Open Work Orders and All Open Requests (Status < 950) to get all records, then let the drill downs do the sorting. (Say by Department, then Supervisor) Works well for a manager type (City Manager, etc).

Example # 9

Find all Work Orders that do not have an Asset

`WKORDER WHERE WO_ID NOT IN (SELECT AS_WO_ID FROM WKWOASSET)`

Notes: _____
