Table of Contents

[Introduction 3](#_Toc442699378)

[Objectives 3](#_Toc442699379)

[WIMS Database Overview 3](#_Toc442699380)

[Basic WIMS Tables 5](#_Toc442699381)

[Basic SQL Queries 7](#_Toc442699382)

[Example 1: List all variables in my database 7](#_Toc442699383)

[Example 2: List all variables with units of MGD 7](#_Toc442699384)

[Example 3: List all variables with units of MGD or mg/L 8](#_Toc442699385)

[Example 4: Show me variables that do not have SCADATAGS 8](#_Toc442699386)

[Example 5: Put the list in VarNum order 9](#_Toc442699387)

[Example 6: Sort the list by units then VarNum 9](#_Toc442699388)

[Example 7: I only want to see VarNum, Name, Units, and Varid 10](#_Toc442699389)

[Example 8: I only want VarNums under 100 with units of mg/L 10](#_Toc442699390)

[Example 9: Find all variables in the list that where the name contains SS. 10](#_Toc442699391)

[Example 10: Find all variables that contain SS but not VSS 11](#_Toc442699392)

[Example 11: I want to list all calculated variables 12](#_Toc442699393)

[Example 12: Show daily data edited since yesterday 13](#_Toc442699394)

[Example 13: Show daily data edited since yesterday where yesterday is calculated 13](#_Toc442699395)

[Example 14: How many records were updated since yesterday 14](#_Toc442699396)

[Example 15: How many records were edited by user 15](#_Toc442699397)

[Example 16: Only show users that edited more than 5 records 15](#_Toc442699398)

[Example 17: I want to see the last 5 edited records from datatbl 16](#_Toc442699399)

[QUIZ: 17](#_Toc442699400)

[Update Queries 18](#_Toc442699401)

[Replace 'Influent' with 'Raw' in Variable Names. 20](#_Toc442699402)

[Replace a number in an equation 20](#_Toc442699403)

[Use concatenation to update a field 20](#_Toc442699404)

[Delete Queries 21](#_Toc442699405)

[Insert Queries 22](#_Toc442699406)

[SQL Tools 22](#_Toc442699407)

[Join Queries 23](#_Toc442699408)

[Inner Join 23](#_Toc442699409)

[Left Outer Join 26](#_Toc442699410)

[Example: Show me Variable Num, Name, Location name, and units 26](#_Toc442699411)

[Example: Multiple left outer joins on the Sample Table. 27](#_Toc442699412)

[Example: Display Sample Status description with sample 28](#_Toc442699413)

[Other Joins 29](#_Toc442699414)

[SQL Versions: 30](#_Toc442699415)

[Example: Find version used to create a backup 30](#_Toc442699416)

[Interfaces 31](#_Toc442699417)

[Using SQL in WIMS 31](#_Toc442699418)

[Lesson 1: Simple Test List report: 34](#_Toc442699419)

[Lesson 2: Sample Order Template 38](#_Toc442699420)

[Example: Get Data Directly from External Source 40](#_Toc442699421)

[Import from External SQL Databases 42](#_Toc442699422)

[Dashboard button to execute SQL 42](#_Toc442699423)

[Useful Links 45](#_Toc442699424)

[Example Queries: 46](#_Toc442699425)

[DR3900 tests run report: 46](#_Toc442699426)

[Pennsylvania SDWA 1 47](#_Toc442699427)

[DMR data change report 48](#_Toc442699428)

[LIMS Queries 49](#_Toc442699429)

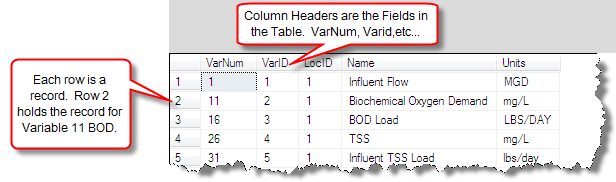
[Updating Variables with SQL console 49](#_Toc442699430)

[Key Definitions 51](#_Toc442699431)

## Introduction

SQL (Structured Query Language), at its simplest, is a basic language that allows you to "talk" to a database and extract useful information. With SQL, you may read, write, and remove information from a database. SQL is standardized and works with a variety of databases including WIMS Databases (Oracle, MS SQL Server, MySQL, Postgres, MS Access).

A database consists of related tables. Data is stored in Tables. Tables are made up of Columns (Fields) and Rows (Records).



### Objectives

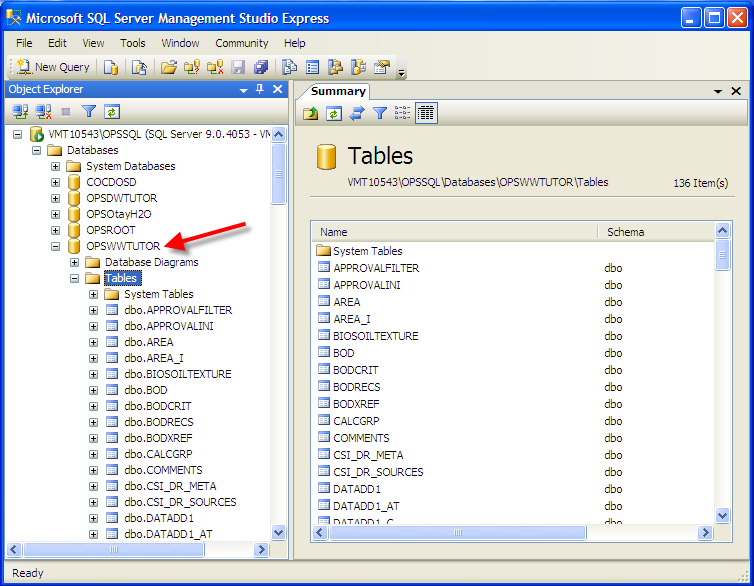
* Understand WIMS database schema
* Perform simple SELECT queries
* Perform SELECT queries with joins
* Learn how to help yourself
* Perform UPDATE queries
* Use SQLFIRST

### WIMS Database Overview

WIMS uses a client application to connect to a Microsoft SQL 2005 or later database server (also support Oracle, however this article covers MS SQL Server examples).  It uses the ADO/OLEDB access technology to connect.

WIMS data is stored in separate MS SQL Server databases under one MS SQL Server.  There is one WIMS Admin database, called OPSROOT that contains a variety of tables including the Facility List, User List, Login History, etc...  This database MUST always exist and is created during install (for WIMS Multi-User) or with Server Setup (for WIMS Multi-User with Database Support).

For each WIMS Facility Database a MS SQL Server Database (catalog) is created and contains the exact same table definitions as well as triggers and stored procedures as other WIMS Facility Databases.  All WIMS Facility Databases start with "OPS".  In the example below, we have 3 facility databases (OPSDWTUTOR, OPSOtayH2O, and OPSWWTUTOR) and OPSROOT.  The Database Owner must be set to OPSDBA for all WIMS databases (OPSROOT and Facility databases).

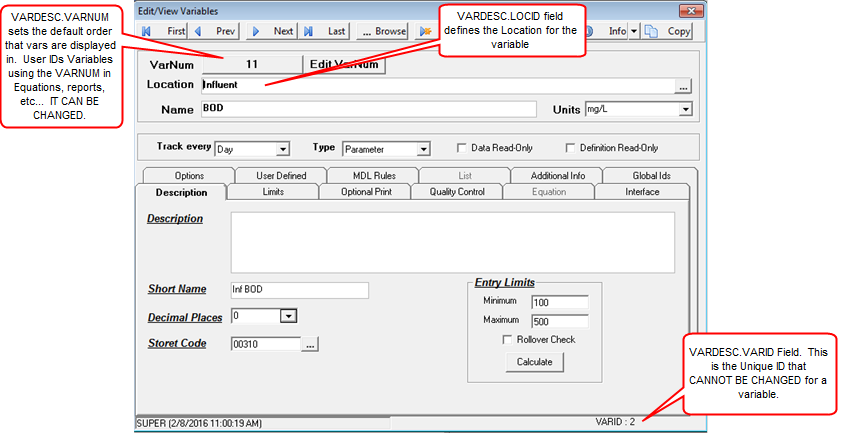


Upon logon, the client application will attempt to connect as user OPSDBA.  The default catalog (database) for this user is OPSROOT.

### Basic WIMS Tables

The most important tables and their descriptions are as follows:

VARDESC – Stores information about all parameters/variables tracked by WIMS.



DATATBL - Tables matching DATA\* are used as data storages for date stamped records. Records are linked back to the VARDESC table via VARID.

DATATBL\_AT – Datatbl audit trail

DATATBL\_I – Additional data for the datatbl.

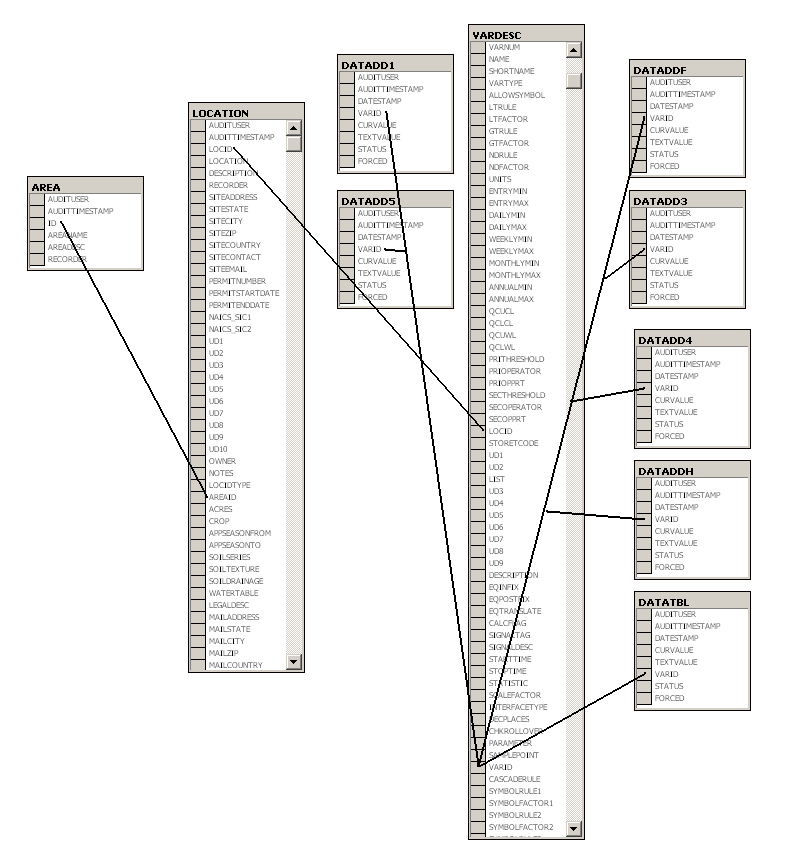
COMMENTS – Result comments for daily vars (datatbl)

Daily detail data (Hourly, 15 Min data, etc…) is stored in the DATADD+ tables. Their structures resemble the one of DATATBL.  For example:  
DATADDH - Stores the hourly frequency data for variables with a VarType of H, N, and B.  
DATADD4 - Stores the 4 hour frequency data for variables with a VarType of 4,G, and E.  
DATADDF - Stores the 15 Minute frequency data for variables with a VarType of F,V, and X.  
DATADD3 - Stores the 30 Minute frequency data for variables with a VarType of 3,W, and Q.  
DATADD5 - Stores the 5 Minute frequency data for variables with a VarType of 5, Y, and R.  
DATADD1 - Stores the 1 Minute frequency data for variables with a VarType of 1, S, and A.  
Daily detail sample comments are stored in the DataDDx\_C where x is the Vartype.  For example:  DATADDH\_C is the sample comments table for hourly data.  
Daily detail audit trail is stored in the DataDDx\_AT table where x is the VarType.  For example: DATADDF\_AT is the audit trail table for 15 Minute data.  
Daily detail additional info is stored in the DataDDx\_I table where x is the VarType.  For example: DATADDF\_I is the additional info table for 15 Minute data.

FILES – Simulates a file system for WIMS clients. Files are stored as Binary Objects.

LOCATION – Contains the list of Locations/sampling points. Used to populate the VarDesc.LocId field. VarDesc.LOCID = Location.LOCID

See <http://www.opssys.com/instantkb/article.aspx?id=13470> for a complete list



## Basic SQL Queries

A query retrieves records from Tables.

SELECT : The SELECT clause specifies the table columns that are retrieved.

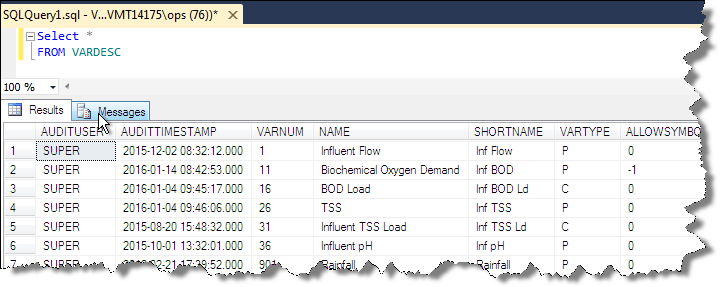
FROM : The FROM clause specifies the tables accessed.

WHERE : The WHERE clause specifies which table rows are used. The WHERE clause is optional; if missing, all table rows are used.

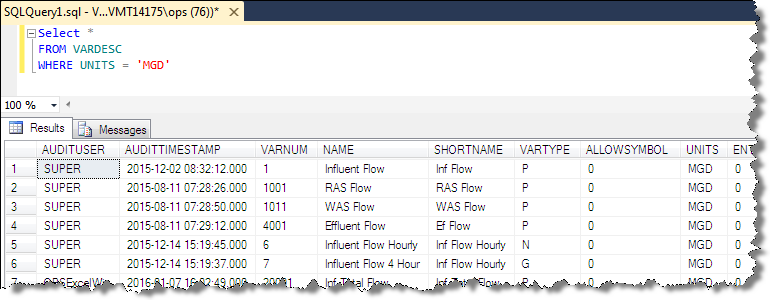
ORDER BY: Sets the sort order of the returned records. Optional

### Example 1: List all variables in my database

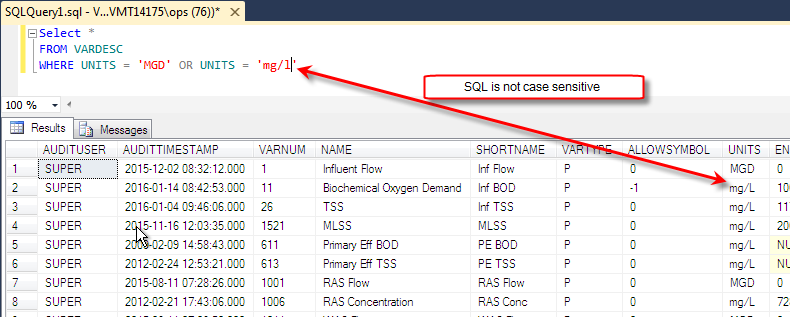
SELECT \*  
FROM VARDESC



### Example 2: List all variables with units of MGD

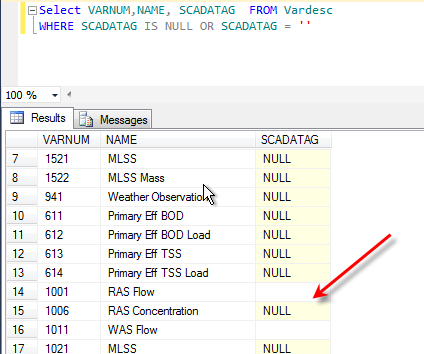


### Example 3: List all variables with units of MGD or mg/L



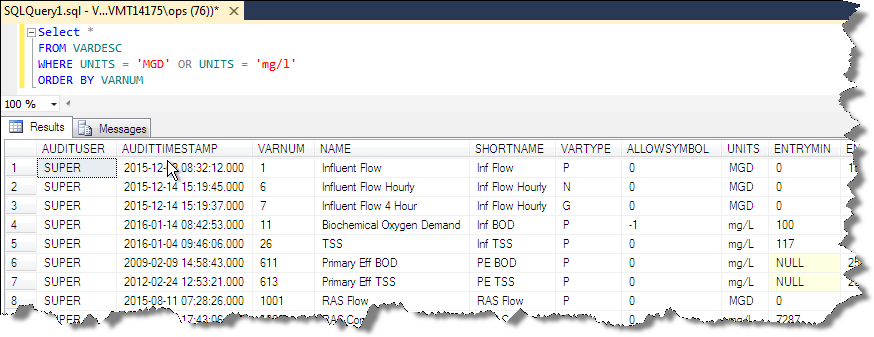
SQL is not case sensitive

### Example 4: Show me variables that do not have SCADATAGS

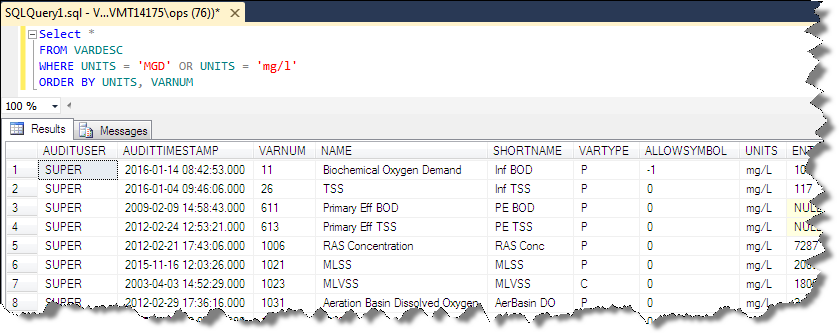


Discussion: Difference between NULL and ‘’

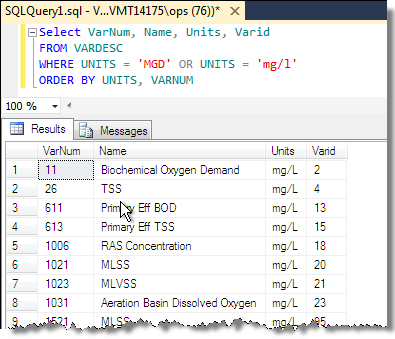
### Example 5: Put the list in VarNum order



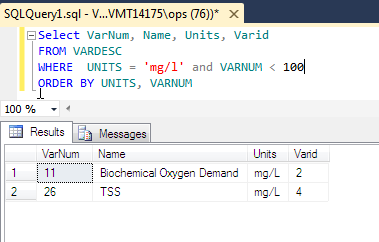
### Example 6: Sort the list by units then VarNum



### Example 7: I only want to see VarNum, Name, Units, and Varid



### Example 8: I only want VarNums under 100 with units of mg/L

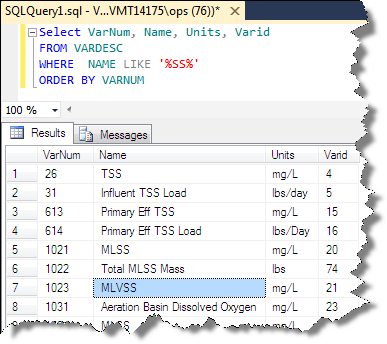


### Example 9: Find all variables in the list that where the name contains SS.

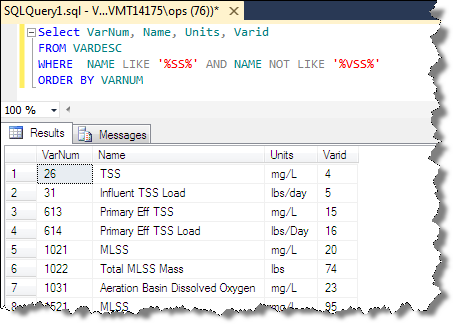
The Wildcard is % in SQL.

% A substitute for zero or more characters

\_ A substitute for a single character



### Example 10: Find all variables that contain SS but not VSS

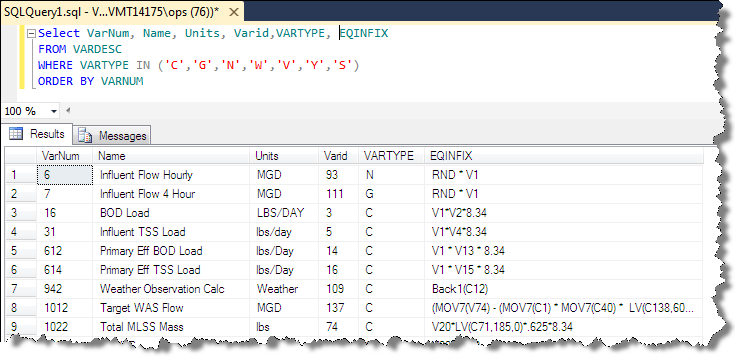


### Example 11: I want to list all calculated variables

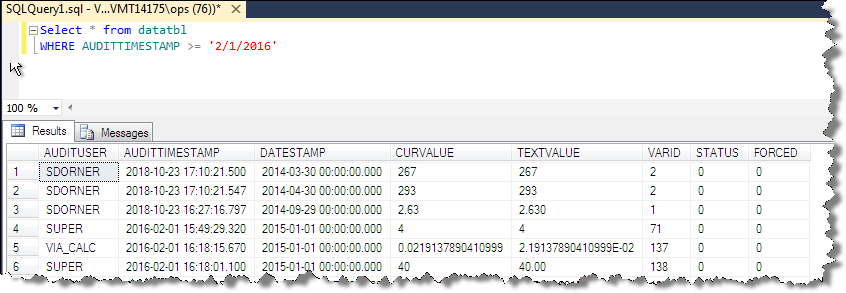
Use the vartype, from KB <http://www.opssys.com/instantkb/article.aspx?id=10592>

* P – Daily variable / parameter
* C – Daily calculated variable
* T – Daily text variable
* 4 – Daily Detail variable tracked every 4 hours
* G – 4 hour calc.
* E – 4 hour text variable
* H – Daily Detail variable tracked every hour
* N – Hourly Calc
* B – Hourly Text
* 3 – Daily Detail variable tracked every 30 minutes
* W – 30 Minute Calc
* Q – 30 Minute Text
* F – Daily Detail variable tracked every 15 minutes
* V – 15 Minute Calc
* X – 15 Minute Text
* 5 – Daily Detail variable tracked every 5 minutes
* Y – 5 Minute Calc
* R – 5 Minute Text
* 1 – Daily Detail variable tracked every minute
* S – Minute Calc
* A – Minute Text

So I want C,G,N,W,V,Y,S. I could do a bunch of ORs, however SQL has an IN clause:



### Example 12: Show daily data edited since yesterday



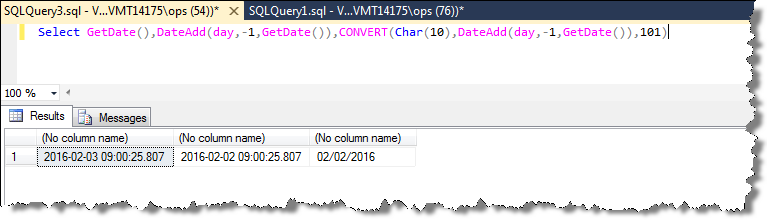
### Example 13: Show daily data edited since yesterday where yesterday is calculated

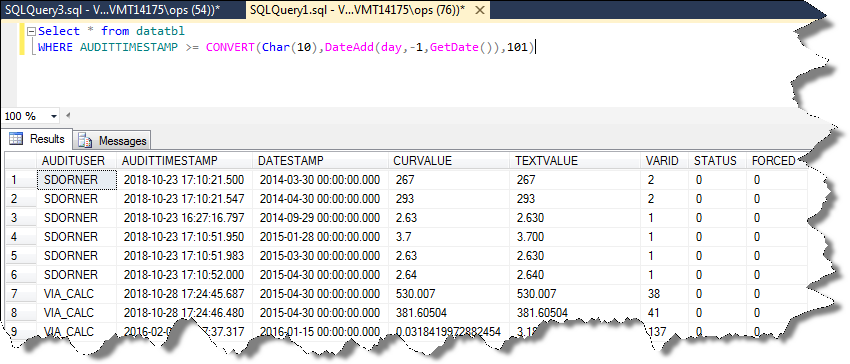
I want to show daily data edited since yesterday but I want to calc the date so I don’t have to change the query every day.

I need to calculate yesterday’s date. So google it. Search for “t sql get yesterday date”. T SQL is Microsoft SQL Server’s variant of SQL (stands for Transact SQL).

GETDATE() – Returns current date time  
DATEADD – Adds or subtracts a number of days, months, etc.. to a date  
CONVERT – Formats values, e.g. dates. Used to cutoff time part.   
DATEDIFF – Returns the specified time (days, months) between dates

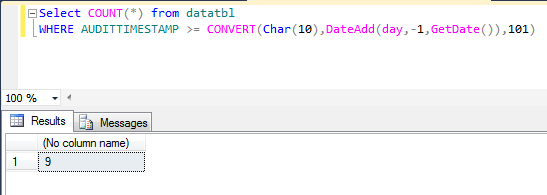
Select GetDate(),DateAdd(day,-1,GetDate()),CONVERT(Char(10),DateAdd(day,-1,GetDate()),101)





### Example 14: How many records were updated since yesterday

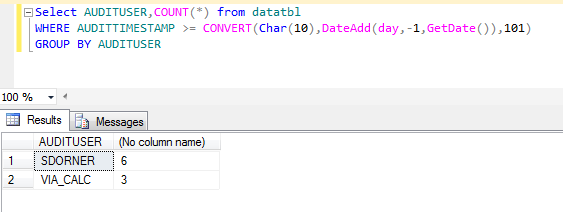
Use the SQL Aggregate function COUNT



See <http://www.w3schools.com/SQL/sql_functions.asp> or google TSQL Aggregate functions.

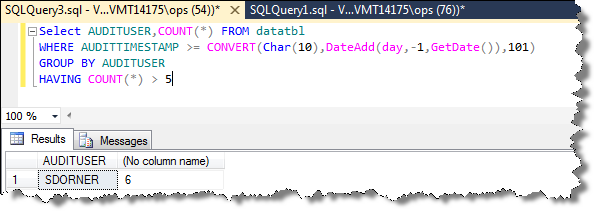
### Example 15: How many records were edited by user

Need to use the Group By Clause

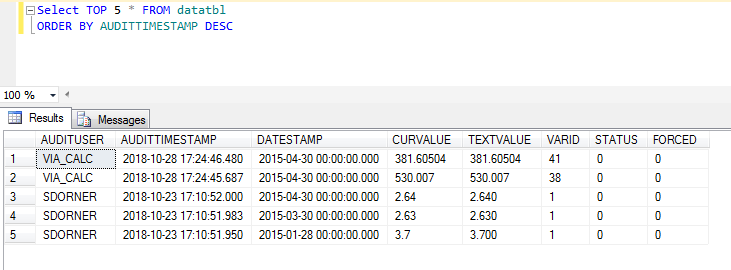


### Example 16: Only show users that edited more than 5 records

Need to use the HAVING clause.



### Example 17: I want to see the last 5 edited records from datatbl



NOTE: ORACLE USES WHERE ROWNUM <=5 for TOP N.

TOP N queries useful when you want to explore data in a table. Example, I have a LIMS database, I would use SELECT TOP 100 \* FROM SAMPLES so I can see some data but don’t want 200K records.

### QUIZ:

1. Show me variables 1000 to 2000 (VarNum)

2. Show me all variables that are linked to a SCADA Interface and SCADATAG is not set

3. Show Yearly Average for V1 - Influent Flow for all years where there is data?

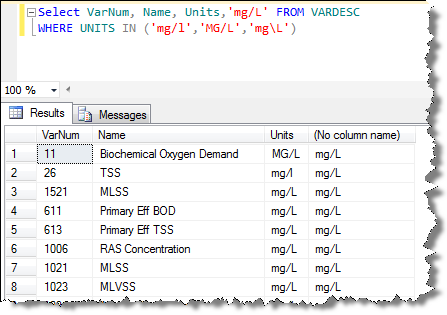
Google “TSQL Get Year from Date” to find a function

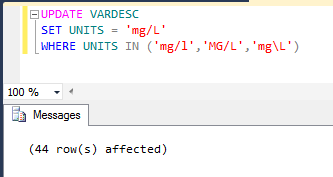
## Update Queries

Used to change existing records in a database:

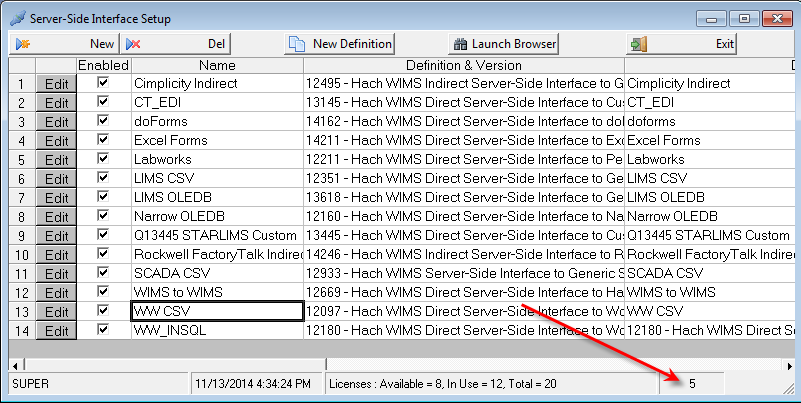
Fix case issues in units – User has units of mg/l, MG/L, mg/L, and mg\L. Set to mg/L to make consistent:

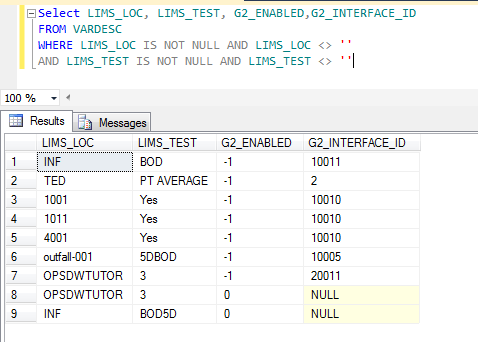
**BE CAREFUL, ALWAYS DO A SELECT QUERY FIRST**





Every var that has a LIMS\_LOC and LIMS\_TEST, set interface to on and set id to 3 (3 is the interface ID of your LIMS Interface). You can find the G2\_INTERFACE\_ID on your Server-Side Interface setup Window.





UPDATE VARDESC

SET G2\_ENABLED=-1, G2\_INTERFACE\_ID=3

WHERE LIMS\_LOC IS NOT NULL AND LIMS\_LOC <> ''

AND LIMS\_TEST IS NOT NULL AND LIMS\_TEST <> ''

### Replace 'Influent' with 'Raw' in Variable Names.

Update Vardesc set name = Replace(name, 'Influent', 'Raw')

### Replace a number in an equation

update vardesc set EQINFIX =replace(cast(EQINFIX as nvarchar),'.012','8.34'), POSTFIX=replace(cast(POSTFIX as nvarchar),'.012','8.34')

Discussion: Why do need CAST. EQINFIX, EQPOSTFIX are ntext fields. nText grows as needed (to fit the data), nvarchar is fixed. However, string functions such as len, replace do not work with ntext, therefore we convert to nvarchar first.

### Use concatenation to update a field

Update the StoretCodes in VARDESC table that do not have zero padding. I.E. StoretCode is 310 instead of 00310:

*update vardesc set StoretCode = '00' + StoretCode where len(StoretCode) = 3*

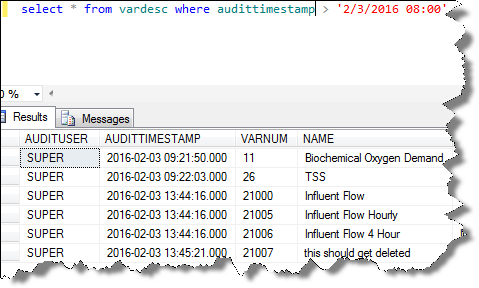
+ is used to concatenate strings in TSQL.

|| is used in Oracle, Oracle also has CONCAT function to increase easy of understanding

## Delete Queries

Delete all variables that have an AuditTimestamp > 2/3/2016 8:00AM

First select the records. I don’t want to delete 11 and 26 as they were edited not added.



DELETE FROM vardesc

WHERE audittimestamp > '2/3/2016 08:00'

AND VARNUM >=21000

## Insert Queries

The INSERT INTO statement is used to add new records in a table.

INSERT INTO comes in two forms, one that specifies the columns and one that does not. If the fields are not specified, you must list the values in their default order (do a Select \* FROM … to get column order).

INSERT INTO table\_name VALUES (value1,value2,value3,...)

INSERT INTO table\_name (column1,column2,column3,...) VALUES (value1,value2,value3,...)

See <http://www.w3schools.com/SQL/sql_insert.asp>

<http://www.opssys.com/InstantKB/Article.aspx?id=14294>

LABCAL\_GENONCEADAY Facility Setting instructs Lab Cal to generate new Samples only once per day. Setting should be used in situations where users have a large number of samples which causes the calendar to take a long time to display.

SQL Server:

INSERT INTO Settings Values ('SUPER',GETDATE(),'LABCAL\_GENONCEADAY','12312014')

Oracle:

INSERT INTO Settings Values ('SUPER',SYSDATE,'LABCAL\_GENONCEADAY','12312014')

## SQL Tools

Microsoft SQL Server Management Studio. Management Studio is a software program from Microsoft that is used for configuring, managing, and administering a Microsoft SQL Server 2012 DBMS. It is recommended that you install Management Studio on the server hosting the WIMS SQL Server 2012 database as it is an invaluable tool for troubleshooting and administration of the WIMS database. This is installed with the WIMS database.

SQL\_CONSOLE – A small Hach WIMS program that allows you to connect to OLEDB Compliant databases and execute SQL

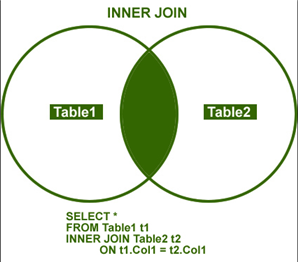
Backup / Restore in SQL Server – DBA Owner, Facility Table

## Join Queries

Joins allow a query to pull data from more than one table. The different types of joins allow us to specify which records to pull from each table.

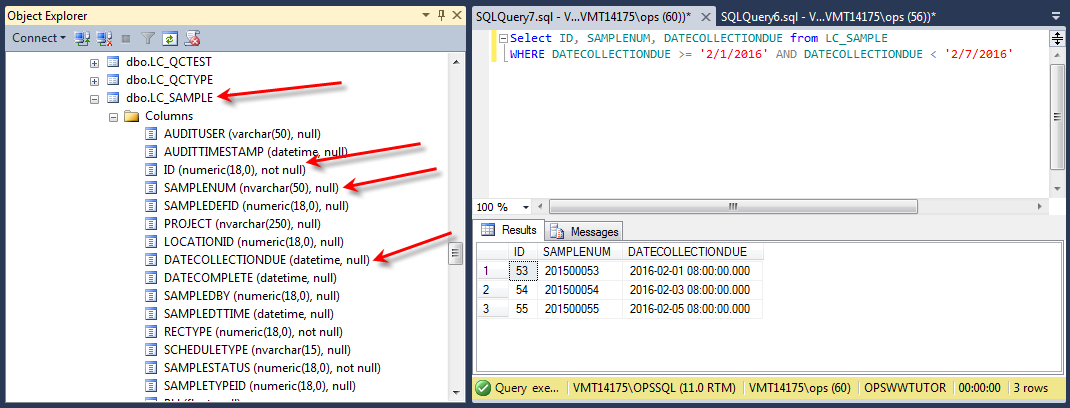
### Inner Join

Inner Join - All rows from both tables as long as there is a match between the columns in both tables. NOTE: If you use a where clause to specify the match you are creating an inner join. This is the most common join type.



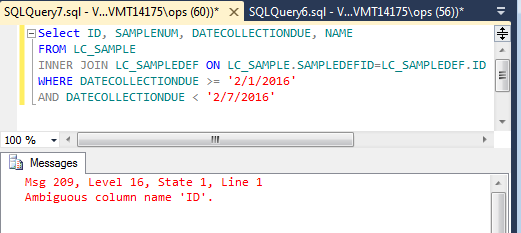
Example: I need to find the Sample Name for all samples that are due this week. The samples are stored in the LC\_SAMPLE table. The Sample Name is not in the LC\_SAMPLE table.

Discussion: WHY NOT?

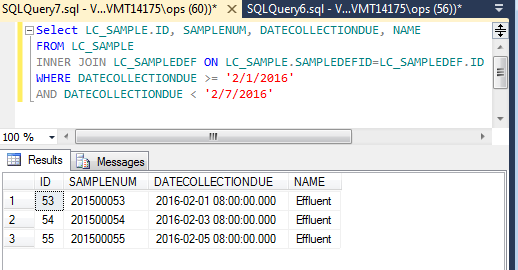


The Sample Name is held in the LC\_SAMPLEDEF table, so I must join the LC\_SAMPLEDEF table.

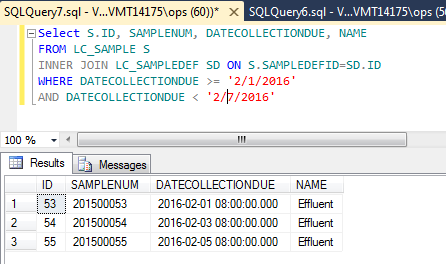
The business rules for Lab Cal mandate that there MUST be an LC\_SAMPLEDEF record for a sample. Therefore we can use an inner join.



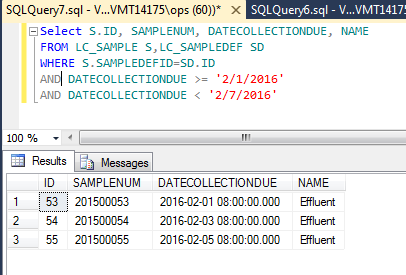
Why did we get Ambiguous column? There is an ID column in both tables. Therefore, we have to tell SQL which table to use.



HINT: Save typing by using Aliases for table names in the query. After referring to the table in the FROM or Join clause put an abbreviation for the table name:



What if I use a WHERE clause to join the tables. That means you are doing an inner join.

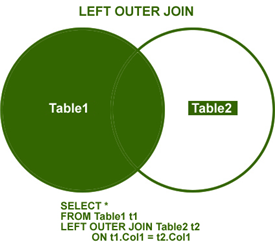


Discussion:

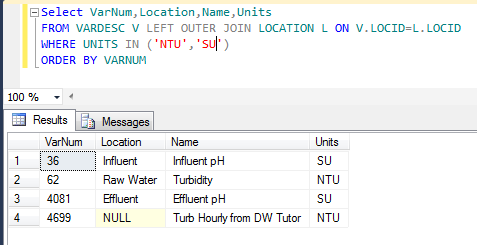
* What records would I get if the Effluent Sample did not exist in the LC\_SAMPLEDEF Table?
* How do I know what fields I should use to join the tables?
  + Fieldname ends in ID
  + Get a database diagram
  + Try it with known data and see

### Left Outer Join

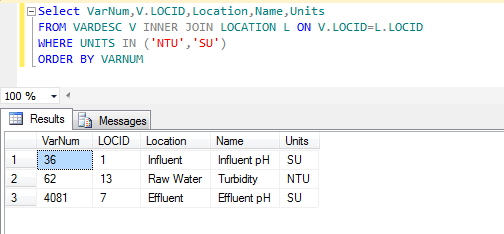
Hint: One way to think about a Left Outer join is as a lookup join. I want to lookup the Location Name for a variable using the LocID field in Vardesc to “lookup” the location, if the Variable is not assigned to a Location (LOCID is null or LOCID does not exist in the Location table) I still want to retrieve the Variable.



### Example: Show me Variable Num, Name, Location name, and units



Notice V4699 is included even though it’s Location (NULL) is not in the Location table. Now change it to an inner join:



V4699 is not included in the result set.

### Example: Multiple left outer joins on the Sample Table.

I want to see the Sample Name, SampleNum, SampledBy, Assigned to, and Location.

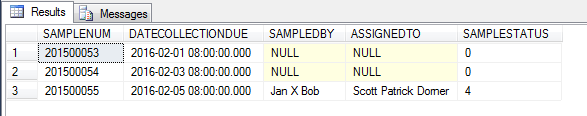
Select SAMPLENUM,DATECOLLECTIONDUE, P1.NAME SAMPLEDBY, P2.NAME ASSIGNEDTO, SAMPLESTATUS

FROM LC\_SAMPLE S

LEFT OUTER JOIN LC\_PEOPLE P1 ON S.SAMPLEDBY=P1.ID

LEFT OUTER JOIN LC\_PEOPLE P2 ON S.ASSIGNEDTOID=P2.ID

WHERE DATECOLLECTIONDUE > '2/1/2016' AND DATECOLLECTIONDUE<'2/6/2016'



SAMPLESTATUS Values:

|  |  |
| --- | --- |
| SAMPLESTATUS | Description |
| 0 | Pending |
| 4 | Received |
| 5 | Analyzed |
| 8 | Skipped |
| 9 | Closed |

### Example: Display Sample Status description with sample

Join the Sample Status Description table. There is not one. GOOGLE TSQL Case statement, Immediate IF (IIF).

Select SAMPLENUM,DATECOLLECTIONDUE, SAMPLESTATUS, SSDESC =

CASE SAMPLESTATUS

WHEN 0 THEN 'Pending'

WHEN 4 THEN 'Rcvd'

WHEN 5 THEN 'Analyzed'

WHEN 8 THEN 'Skipped'

WHEN 9 THEN 'Closed'

END

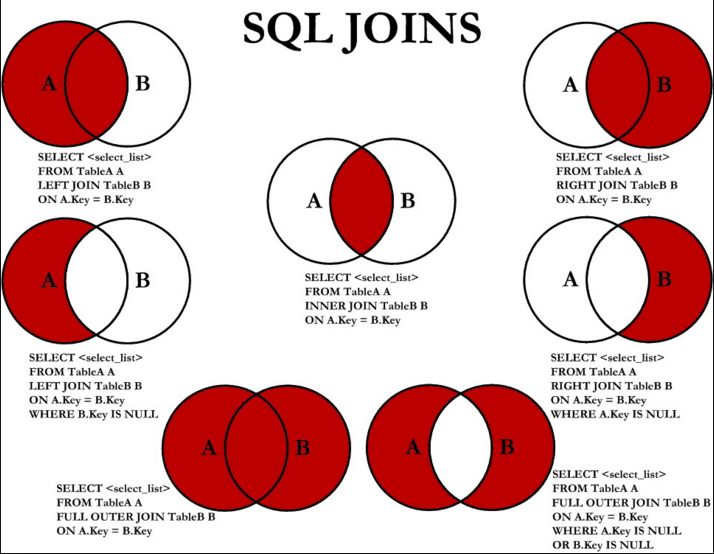
FROM LC\_SAMPLE S

WHERE DATECOLLECTIONDUE > '2/1/2016' AND DATECOLLECTIONDUE<'2/6/2016'



## Other Joins

There are other joins that SQL supports that will not be covered as they are rarely used:



## SQL Versions:

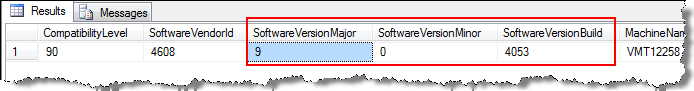


We only support SQL 2005 and later, 2016 has not been released yet.

### Example: Find version used to create a backup

restore headeronly

from disk = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\OPSAUG.BAk'



## Interfaces

Use <http://www.connectionstrings.com/> to find an OLE DB Connection string to use with the interface or SQL\_CONSOLE.

To test connections and see the connection string that is generated use a UDL file. Basic steps:

1. Create a text document

2. Rename extension to .UDL

3. Double Click and Microsoft’s Data Link Properties will be launched.

4. Choose Provider, Connection, etc… and click Test Connection. If it connects. Click OK to Save.

5. Open File in Notepad and you will have the connection string.

.

## Using SQL in WIMS

Spread Report SQLFIRST, SQLRESULT, SQLXFIRST, SQLXRESULT. X versions allow you to pull data from External databases into the report.

Spread reports allows use of SQL statements in the SQLFIRST and SQLRESULT functions to report on the tables that contain the Sample Information, schedule, chain of custody...

Let’s examine a basic SQL query:

SQLFIRST(Col, Row, "Facility","SQL Query", MaxColumns, MaxRows, Parameter1,…,Parameter20)

This query returns a value and the result is put into a Spread Report. If the result of the SQL Query is a table of more than 1 column by 1 row, SQLRESULT must be used to retrieve other values.

Col Column number to retrieve result from

Row Row number to retrieve result from

"Facility" Facility to query against, specified by its unique identifier. If left blank, current facility is used

"SQL Query" SQL Query to execute

MaxColumns Specifies the number of columns in the resulting table

MaxRows Specifies the number of rows in the resulting table

Parameter1..20 OPTIONAL. External values to be used substituted into the query, should the query refer to them. In SQL Query, use @Px@ to refer to the parameter, or #Px# to refer to the parameter as a date, $Px$ as a number.

**REMARKS:** Be aware that SQLFIRST stores each query result table within memory. Use caution when specifying large MaxColumn and MaxRow sizes. Also, restrain from specifying queries that can potentially overwhelm your PC memory.

If you specify 0 for Col and Row parameters, SQLFIRST will perform the query, but it will not return anything. The result table of the query will be accessible by SQLRESULT.

EXAMPLES:

Returns the result located at col 1 and row 1 of the 4 by 30 table of variables for the location entered in Cell A2.

SQLFIRST(1,1,””,"SELECT VarNum,Name, Units,Location from VarDesc where Location = '@P1@'",4,30,A2)

Returns the average for the Varid in A1 for facility OPSSANDY for the dates 1/1/2004 through the date in cell A2.

SQLFIRST(1,1,”OPSSANDY”,"SELECT AVG(CurValue) from DataTbl where VarId = @P1@ and DateStamp >= #P2# and DateStamp <= #P3# ",1,1,A1,"1/1/2004",A2)

Returns the Name from the LC\_People Table for the ID number in cell A2. The query is set to return one result (1 column, 1 Row)

SQLFIRST(1,1,"OPSROCKY", SELECT name from LC\_People where ID=$P1$,1,1,A2)

Returns a value located at Col and Row of an SQL Query result table. This SQL query must be specified using SQLFIRST formula and must be located in Spread Report cell referred to as CellRefference.

SQLRESULT(CellReference, Col, Row)

**Example:**

1. Cell A1 contains the following formula:

SQLFIRST(1,1,””,"select max(curvalue), min(curvalue) from vardesc where varid=1",2,1)

1. Cell A2 contains the following formula:

SQLRESULT(A1,2,1)

The SQLRESULT refers to cell A1, which runs the query specified by SQLFIRST. The query is set to return 2 results (maximum and minimum). SQLFIRST in A1 only returns the maximum results. Hence SQLRESULT is A2 returns the minimum result.

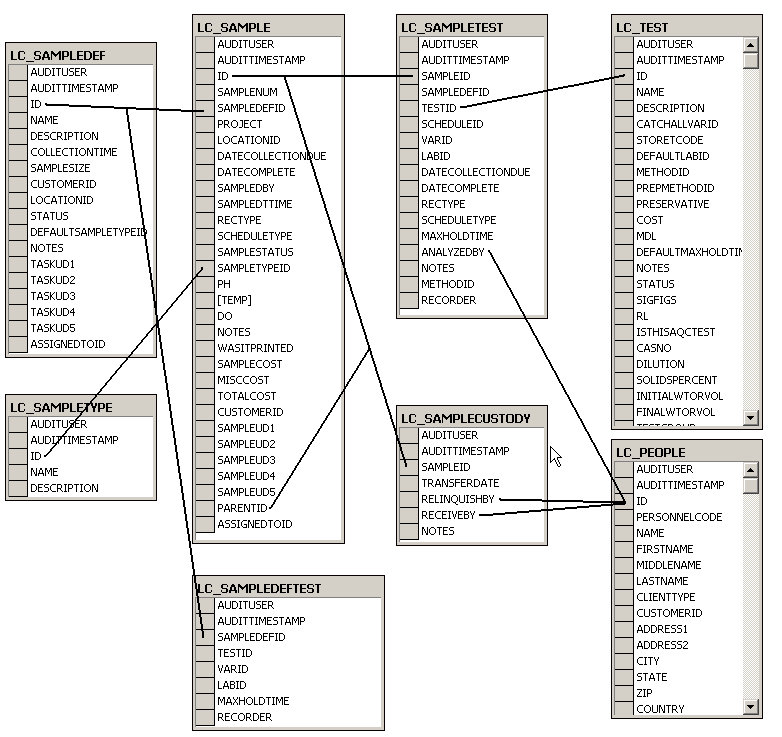
1. Cell B1 contains the following formula:

SQLFIRST(1,1,””,"select varnum,name,units from vardesc where varid=1",2,1)

1. Cell B2 contains the following formula:

SQLRESULT(B1,"UNITS",1)

The SQL Result refers to cell B1, which runs the query specified by SQLFIRST and returns the units, or other specified info from the SQLFIRST query, where varid=1



Key *Lab Cal* Tables

LC\_CUSTODYNOTES

LC\_METHOD List of Methods.

LC\_PEOPLE List of Personnel

LC\_QCTEST

LC\_QCTYPE

LC\_SAMPLE The sample information table for all scheduled and closed samples.

LC\_SAMPLECUSTODY The Chain of Custody records for a sample. One to Many relationship to LC\_SAMPLE, LC\_SAMPLECUSTODY.SAMPLEID = LC\_SAMPLE.ID

LC\_SAMPLEDEF The Sample Definitions.

LC\_SAMPLEDEFTEST The test(s) assigned to a Sample Definition.

LC\_SAMPLETEST The test information (ie analyst, when run…) for each test in a sample (one to many to LC\_Sample, LC\_SAMPLETEST.SAMPLEID = LC\_SAMPLE.ID). NOTE: Results are stored in the WIMS variable. Therefore use the V or VT Spread Function to get the result.

LC\_SAMPLETYPE The list of Sample Types.

LC\_TEST List of Tests.

This was just a brief introduction to SQL. For more information there are many books for SQL for beginners, or search the Internet for SQL tutorials.

### Lesson 1: Simple Test List report:

We want to develop a report the list the tests that the lab performs:

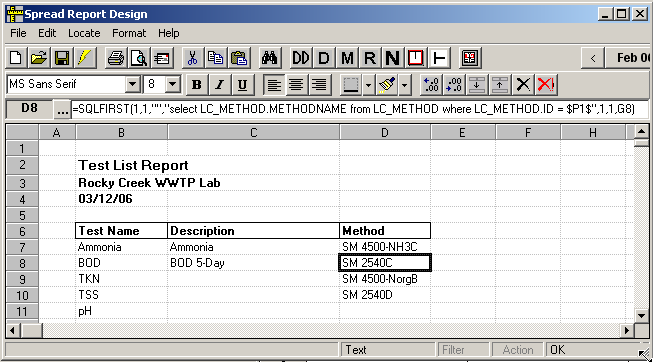


Figure 49

1. Get into Spread Design. Go to *Design, Spread Reports*.
2. Click on B7 – this is the cell where we want the body of the report to start.
3. Now Go to *Locate, SQL Results*

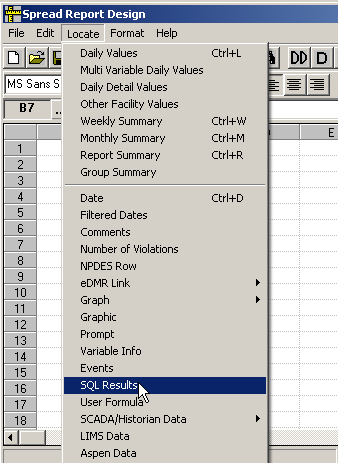
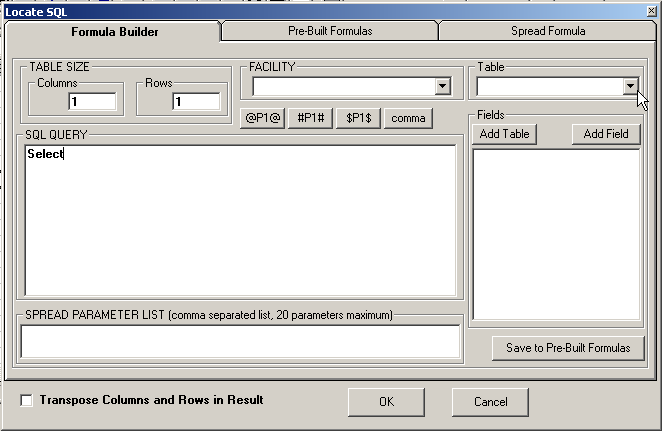
 

Figure 50

1. Select the LC\_Tests table and pick the fields you want displayed. Set the FROM clause and use the ORDER BY clause to sort the results. Click **OK** to locate the results:

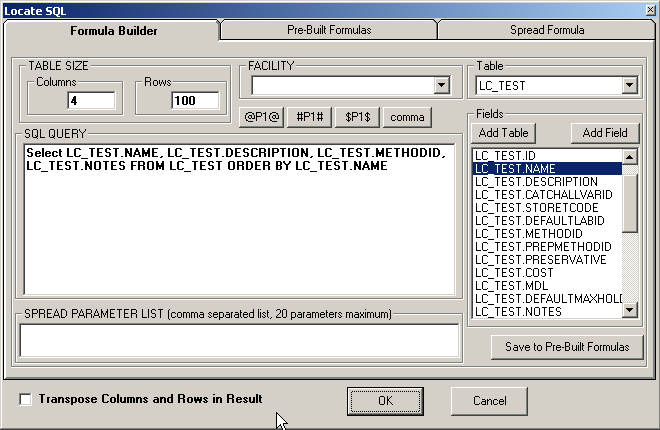


Figure 51

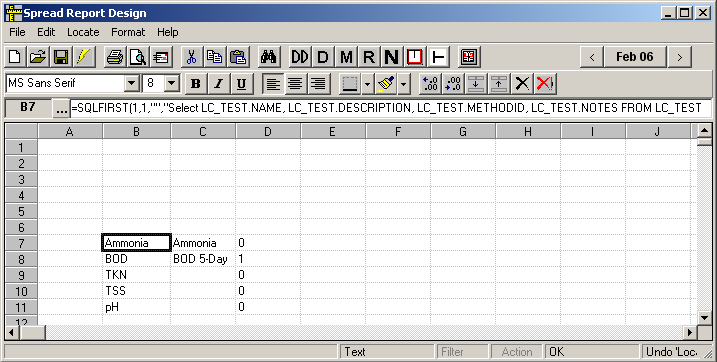
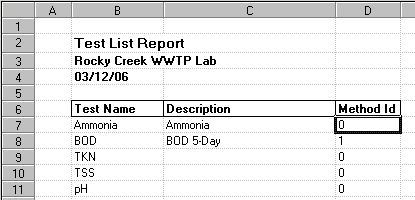


Figure 52

1. Enter the titles and set column widths as shown below.
2. Now we want to get the Method Name instead of the Method Id. To get the Method Name we have to lookup the MethodID in the LC\_Method table.
   1. Move D6:D106 to G6:G106
   2. Click on D7 and go to *Locate, SQL Results*.



***TIP****: To get the current date, use Locate, Dates*

Figure 53

1. Write your select query to change the Method ID to the Method Name.

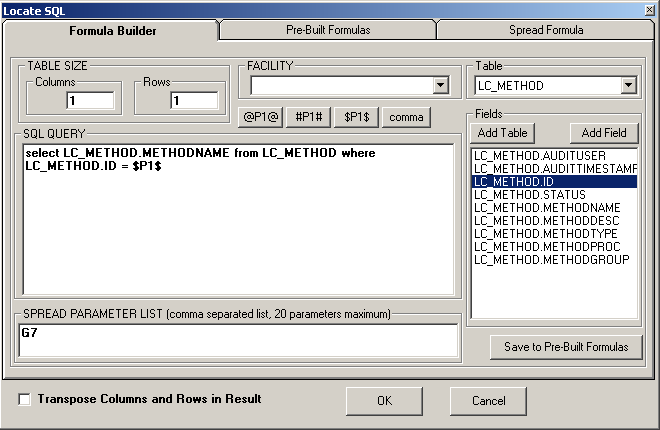


Figure 54

***TIP****: use F5 Copy Special, F6 Paste Special*

1. Copy the formula in cell D7 to D8:D106.
2. Set the column header in D6 to “Method”
3. Click on Cell B7 and go to Format, Freeze Columns and Rows. This will repeat the header information on each page when printing the report.
4. Hide Column G: Click on any cell in column G and go to Format, Col, Hide.
5. That’s it. Save the report.

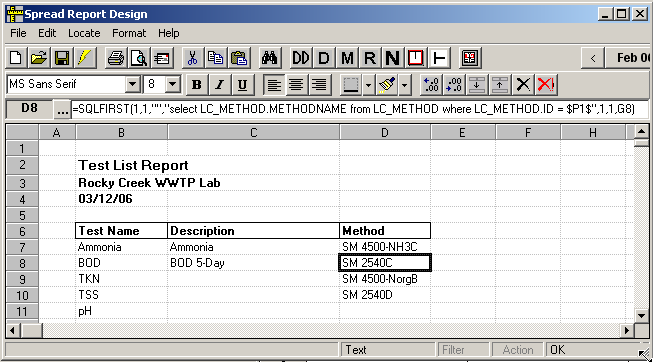


Figure 55

### Lesson 2: Sample Order Template

The Sample Order Report displays sample information including Tests, Chain of Custody, and sample location and time.

1. Get into Spread Design and go to *File, New*. Choose the Lab Cal Sample Order.ss3 template from the list.

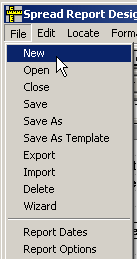
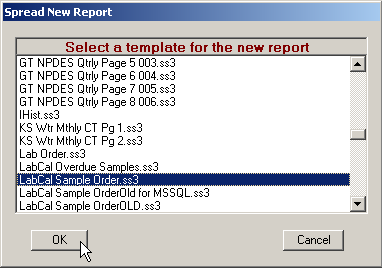
 

Figure 56

1. Click on cell K7

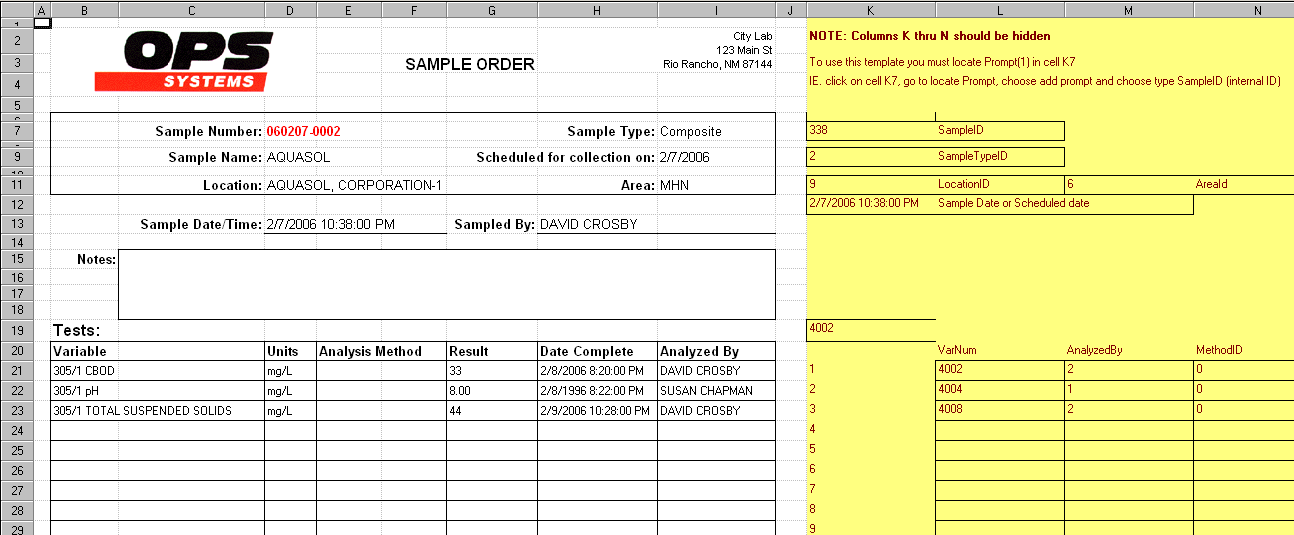


Figure 57

*(Notice columns K though N are yellow. These columns should be hidden. Go to Highlight columns K through N and go to Format, Column, Hide).*

1. Locate a SampleID prompt:

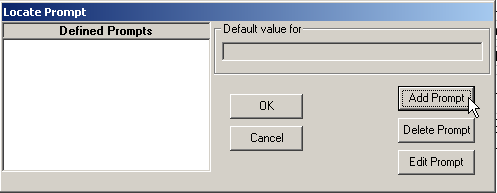
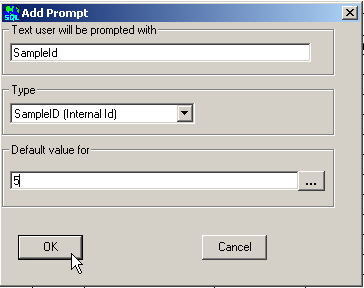
 

Figure 58

You are now ready to print your report. When using the Lab Calendar, the print button runs a report with a special report type of Sample Order. To identify this report as the one you want printed when the Lab Calendar Print button is clicked,

1. Go to *File, Report Options*. Set the Report Type to ‘Sample Order’ and click **OK**.

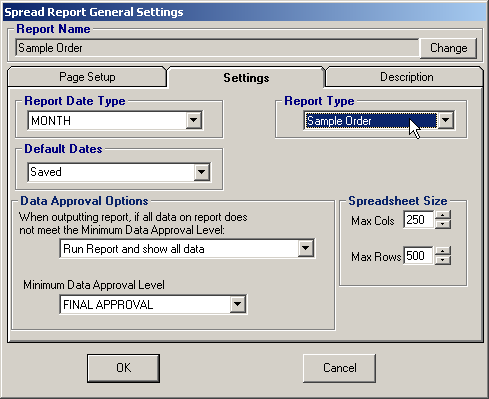


Figure 59

1. Go to *File, Save* and Save the Report as ‘Sample Order’.

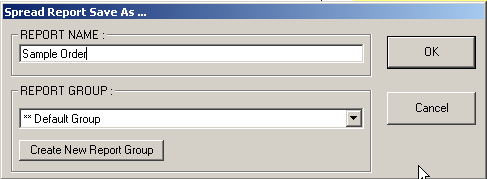


Figure 60

1. You can now print Sample Orders using the Print button in Lab Cal.

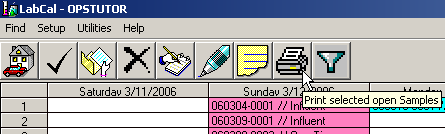


Figure 61

### Example: Get Data Directly from External Source

Customer needs the to count number of times “Chlorine Residual, Total” was analyzed by the lab has this month

We have in the WIMS database all the results that need to be reported but not the QC. Also, they want to know by Analysis Date (we track sample date). They want this on the dashboard.

Do an aggregate COUNT query directly against the LIMS Database. Example uses an Aspen LIMS database.

First develop query in Management studio:

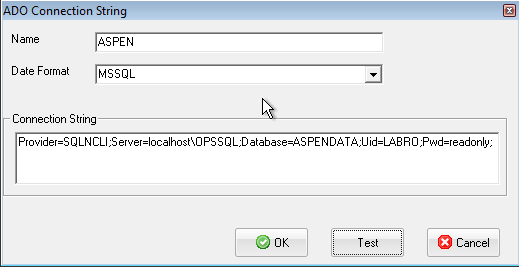
Select COUNT(\*)

FROM TESTS

WHERE TESTNAME = 'Chlorine Residual, Total'

AND ANALYSISDATE >= '4/1/2015' AND ANALYSISDATE < '5/1/2015'

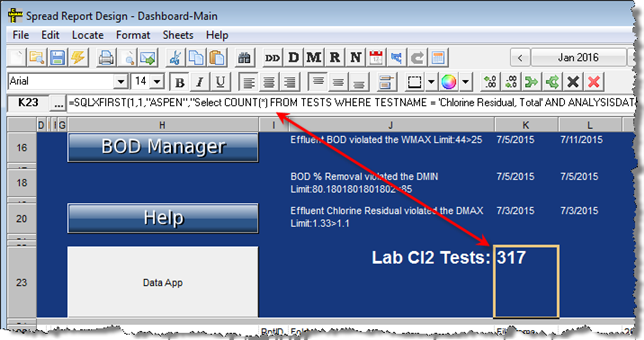
First get into External Data Source Setup and setup a connection to your LIMS DB:



Now get into spread design in your dashboard:

Enter in the following equation:

=SQLXFIRST(1,1,"ASPEN","Select COUNT(\*) FROM TESTS WHERE TESTNAME = 'Chlorine Residual, Total' AND ANALYSISDATE >= '4/1/2015' AND ANALYSISDATE < '5/1/2015'",1,1)

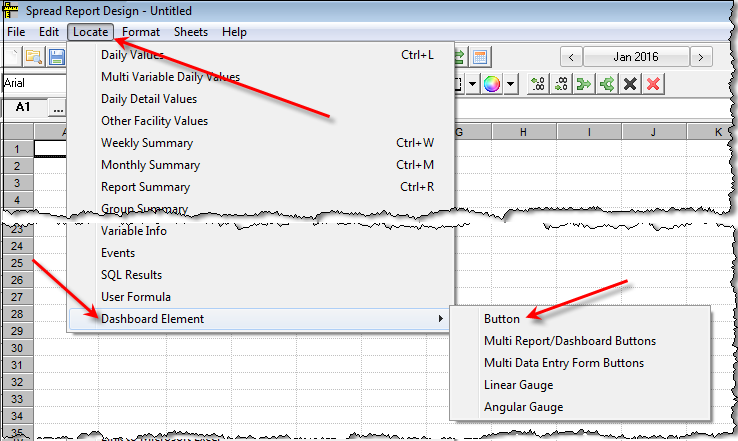


### Import from External SQL Databases

Get the count as a Variable.

### Dashboard button to execute SQL

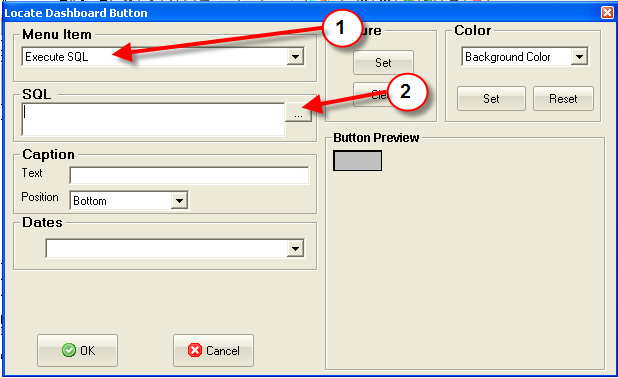
In spread design, locate a dashboard button:



Example: Show all records for variable 1 in the specified date range

1. Choose Execute SQL

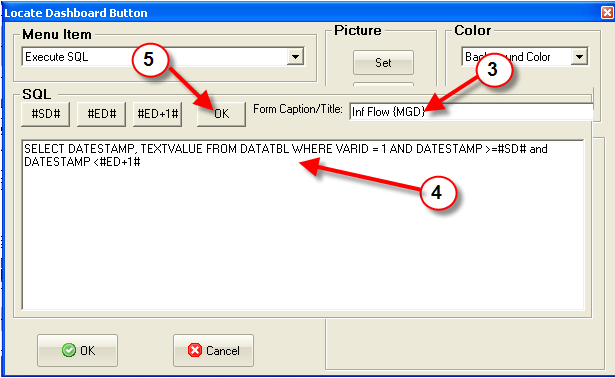
2. Click ... to expand the SQL Statement input box.



3. Enter the title for the Display Results Form.

4. Enter the SQL Statement. SELECT DATESTAMP, TEXTVALUE FROM DATATBL WHERE VARID = 1 AND DATESTAMP >=#SD# and DATESTAMP <#ED+1#

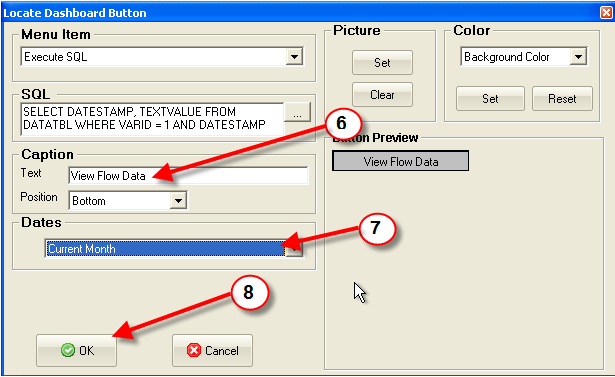
5. Click OK to collapse the SQL input box.



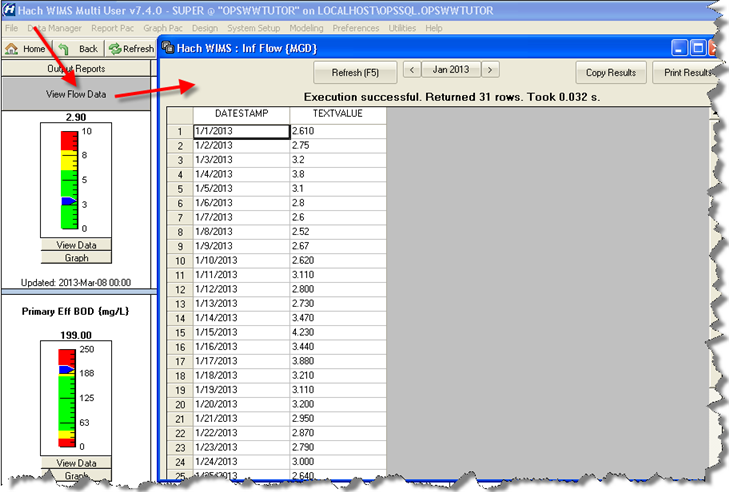
6. Set the button caption

7. Set the Date Range for the query

8. Click OK to locate the button on the report.



On the dashboard, click the button and the results are displayed:



<http://www.opssys.com/instantkb/article.aspx?id=14072>

## Useful Links

SQL Online Tutorial: <http://www.w3schools.com/SQL/>

Example MS SQL Queries: <http://www.opssys.com/instantkb/article.aspx?id=10625>

Server Migration: <http://www.opssys.com/InstantKB/article.aspx?id=14181>

Citect Interface with Q12160: <http://www.opssys.com/InstantKB/article.aspx?id=13957>

SQL Views: <http://www.opssys.com/instantkb/article.aspx?id=14185>

Escondido: <http://www.opssys.com/instantkb/article.aspx?id=14187>

## Example Queries:

### DR3900 tests run report:

SELECT datestamp, sampleid AS location, m.name, curvalue, unit, dilution, program, lotnumber, dateofexpiry, error, comment

FROM csi\_dr\_meta m JOIN

(SELECT datestamp, curvalue FROM vardesc v JOIN ((SELECT varid, datestamp, curvalue FROM dataddH

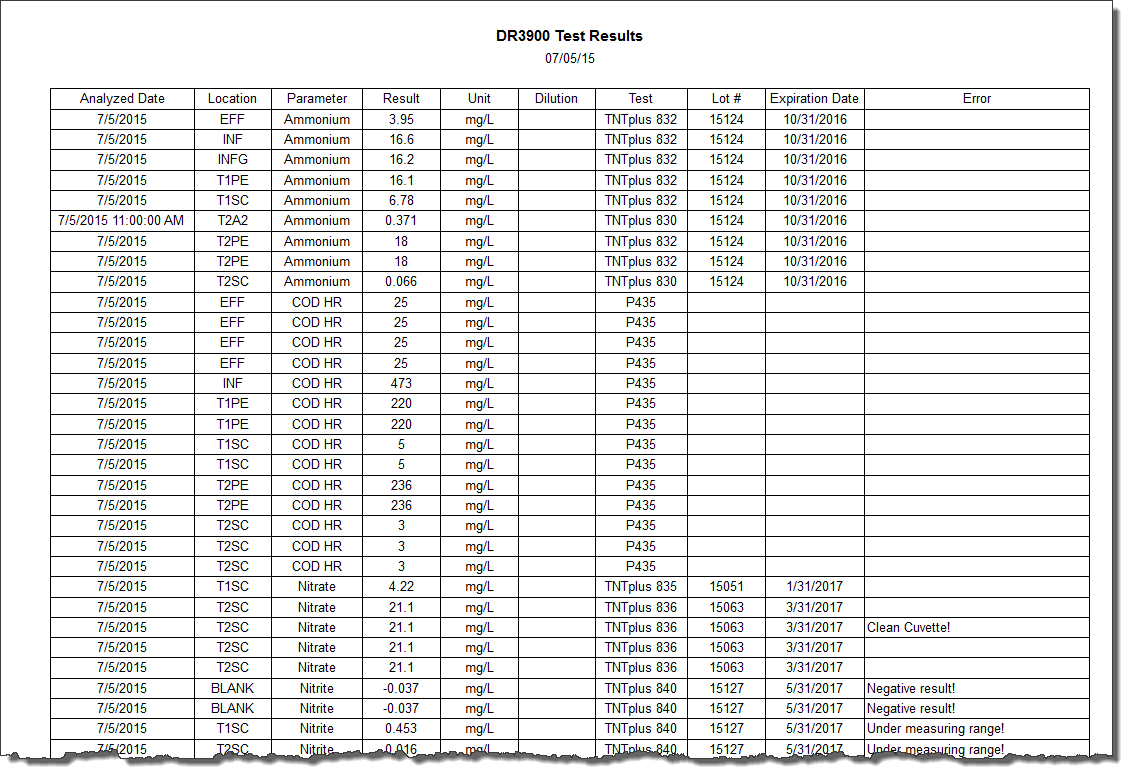
UNION

SELECT varid, datestamp, curvalue FROM datatbl)

WHERE varid IN (SELECT varid FROM vardesc WHERE source\_enabled =-1) AND audituser = 'OPSSYS\_BD') AS data ON v.varid=data.varid

WHERE datestamp >= #P1# AND datestamp < #P2#) AS pd ON curvalue=result

ORDER BY datestamp, location, name



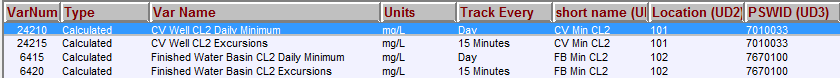
### Pennsylvania SDWA 1

* Requires daily minimum CL2 residual unless 15 minute minimum drops below limit. If below limit, must report first 15 minute value below and 15 minute value when no longer below minimum.

SELECT v.UD3, v.VARNUM, v.VARID,curvalue, f.datestamp,v.UD4,v.UD2 FROM VARDESC v JOIN dataddf f on v.varid=f.varid

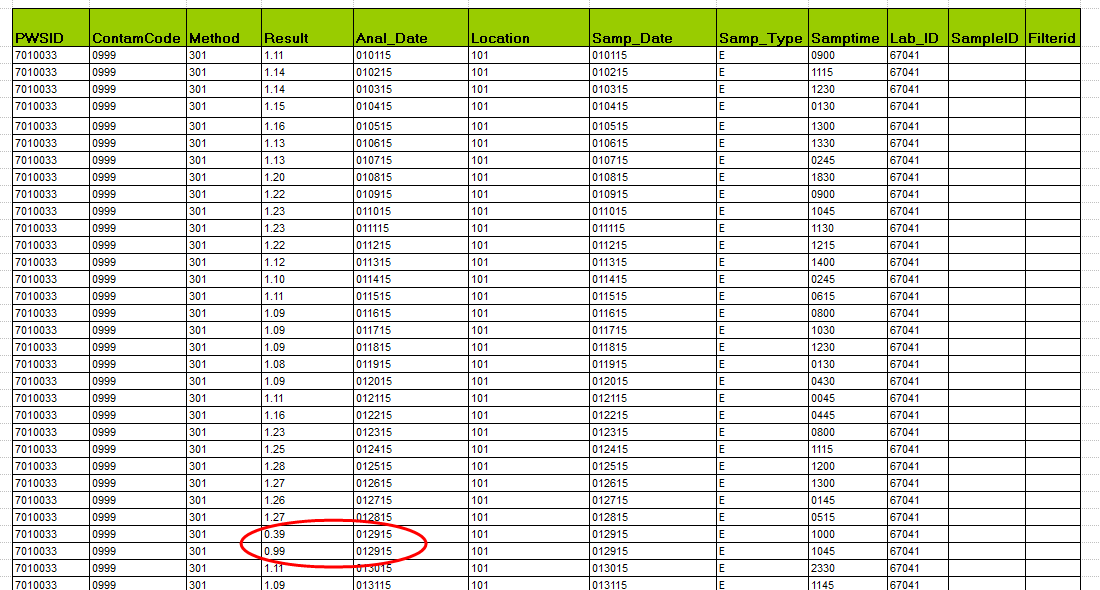
WHERE UD1 LIKE '%Min Cl2' and f.datestamp>=#P1# and f.datestamp<=#P2# UNION SELECT v.UD3,v.VARNUM,v.VARID,curvalue, d.datestamp,v.UD4,v.UD2 FROM VARDESC v JOIN datatbl d on v.varid=d.varid

WHERE UD1 LIKE '%Min Cl2' and d.datestamp>=#P1# and d.datestamp<=#P2# order by v.ud3, v.UD2, f.datestamp



V24210= IF( DDMIN( C24200,0,23) < 0.5, BLANK, IF( isblank(DDMIN( C24200,0,23)), -1, DDMIN(C24200,0,23)))

V24215= IF ( V24200 < 0.5 and back1(C24200) > 0.5, V24200, IF( V24200 < 0.5 and back1(C24200) < 0.5, blank, IF( back1(C24200) < 0.5 and V24200 > 0.5, V24200, BLANK)))



### DMR data change report

select a.audittimestamp, a.datestamp, a.audituser, l.varnum, l.name, a.curvalue, a.oldvalue

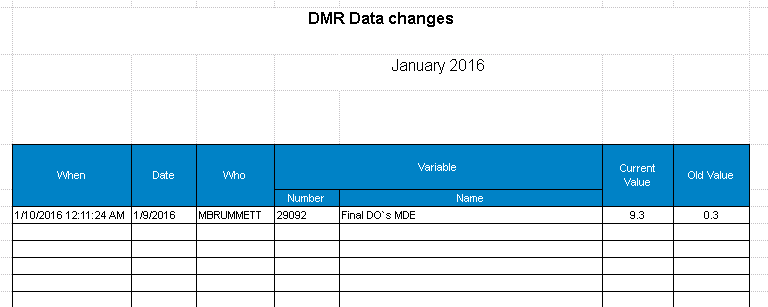
from(select varid, varnum, name from vardesc where varid in (select varid from mdevars where ID = 159 and varid>0)) l

join datatbl\_at a on l.varid=a.varid

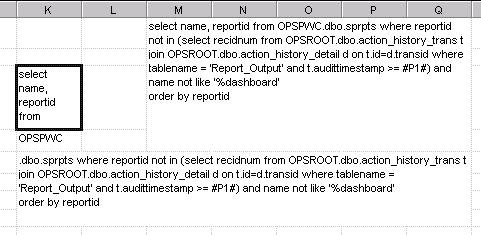
where a.audituser not like 'VIA\_%'

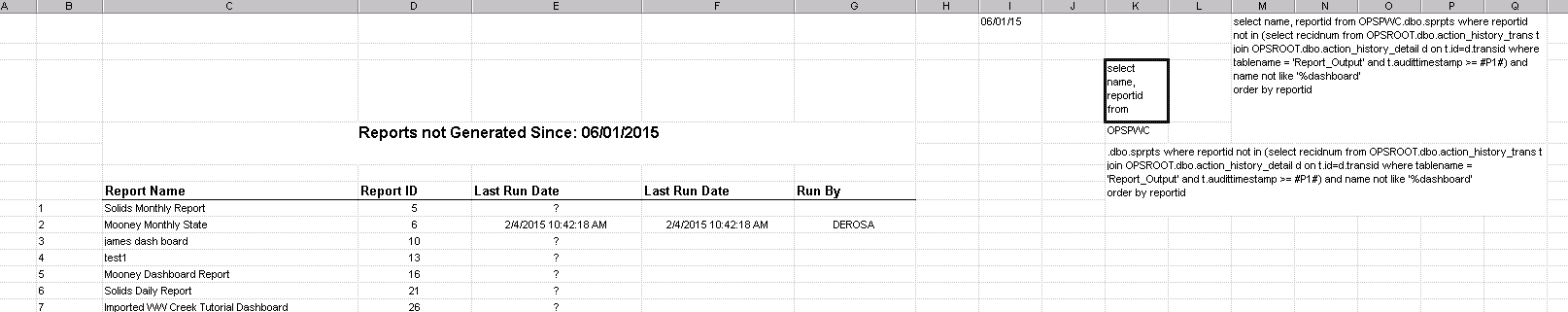
and action = 2

and datestamp >= #P1# and datestamp <= #P2#



Reports not run since … Report





### LIMS Queries

\*\*\*\*\*\*\*\*\*\*\*ATL (Oracle)\*\*\*\*\*\*\*\*\*\*\*

SELECT o.site as LOCCode, r.param as ACode, o.matrix as matrix, r.result as Result,

o.collectdate as COLDate, r.Qualifier as DataQualifier, r.samplenumber as SampleNumber,

r.units as Units

FROM SMSU.results r, SMSU.orderdetails o

WHERE o.samplenumber=r.samplenumber AND r.resultstatus=3 AND r.result is not null

AND o.site is not null

AND r.approveddate> TO\_DATE('27-May-2015 12:00:00 AM', 'dd-Mon-yyyy HH:MI:SS AM')

AND r.approveddate< TO\_DATE('30-Jun-2015 08:01:34 AM', 'dd-Mon-yyyy HH:MI:SS AM')

ORDER BY LOCCODE, o.collectdate

\*\*\*\*\*\*\*\*\*\*\*ATL (MSSQL)\*\*\*\*\*\*\*\*\*\*\*

SELECT o.site as LOCCode, r.param as ACode, o.matrix as matrix, r.result as Result,

o.collectdate as COLDate, r.Qualifier as DataQualifier, r.samplenumber as SampleNumber,

r.units as Units

FROM SMSU.results r, SMSU.orderdetails o

WHERE o.samplenumber=r.samplenumber and r.resultstatus=3 and r.result is not null

and o.site is not null AND r.approveddate> #SD# and r.approveddate< #ED#

ORDER BY LOCCODE, o.collectdate

\*\*\*\*\*\*Generic used with ATL (MSSQL) including CustomerID\*\*\*\*\*\*\*\*

SELECT o.CustomerID+'-'+o.site as SAMPLOC, r.param as ANALYTE, r.result as FINAL,

o.collectdate as SAMPDATE, r.Qualifier as DataQualifier, r.Commnt as ResultComment,

r.samplenumber as SampleNum, r.units as Units

FROM SMSU.results r, SMSU.orderdetails o

WHERE o.samplenumber=r.samplenumber and r.resultstatus=3 and r.result is not null and o.site is not null AND r.approveddate> #SD# and r.approveddate< #ED# ORDER BY SAMPLOC, o.collectdate

\*\*\*Generic used with ATL (MSSQL) using start of collection instead of collection date field\*\*\*\*

SELECT o.site as SAMPLOC, r.param+'-'+o.matrix as ANALYTE, o.matrix as matrix,

coalesce(r.qualifier,'')+r.result as FINAL, o.Startcollectdate as SAMPDATE, o.collectdate as EndColDate, r.Qualifier as DataQualifier, coalesce(r.qualifier,''), r.samplenumber as SampleNum,

r.units as Units

FROM SMSU.results r, SMSU.orderdetails o

WHERE o.samplenumber=r.samplenumber and r.resultstatus=3 and r.result is not null and o.site is not null AND r.approveddate> #SD# and r.approveddate< #ED# ORDER BY SAMPLOC, o.Startcollectdate

### Updating Variables with SQL console

Change SCADA tag to upper case

Update vardesc set scadatag = upper(scadatag) where scadatag is not null

Replace a number in an equation

update vardesc set EQINFIX =replace(cast(EQINFIX as nvarchar),'.012','8.34') where . . .

then do the same for EQPostFix

Change the week definition

update vardesc set eqinfix = replace(cast(eqinfix as nvarchar),'WAVG7','WAVG1'), eqpostfix = replace(cast(eqpostfix as nvarchar),'WAVG7','WAVG1') where eqinfix like '%wavg%'

then do the same for EQPostFix

Limits search

select varnum, v.name, l.description, l.startdate, l.enddate from limits l left outer join vardesc v on v.varid= l.varid where l.enddate = '12/31/2011' order by varnum asc

## Key Definitions

Tables: Data is stored in Tables. Tables are made up of Columns (Fields) and Rows (Records).

Views: A virtual table based on an SQL Query. Acts just like a table.

Indexes: An index can be created in a table to find data more quickly and efficiently. **Note:** Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So you should only create indexes on columns (and tables) that will be frequently searched against.

Stored Procedures – A set of SQL Statements that can be executed to perform an operation. WIMS databases have a stored procedure

Triggers – A type of Stored Procedure that executes every time an action such as a update, insert, delete occurs on a table.

Logins/Security

A schema is a collection of database objects (as far as this hour is concerned—tables) associated with one particular database username. This username is called the schema owner, or the owner of the related group of objects. You may have one or multiple schemas in a database. Basically, any user who creates an object has just created his or her own schema. So, based on a user's privileges within the database, the user has control over objects that are created, manipulated, and deleted. A schema can consist of a single table and has no limits to the number of objects that it may contain, unless restricted by a specific database implementation.

Say you have been issued a database username and password by the database administrator. Your username is USER1. Suppose you log on to the database and then create a table called EMPLOYEE\_TBL. According to the database, your table's actual name is USER1.EMPLOYEE\_TBL. The schema name for that table is USER1, which is also the owner of that table. You have just created the first table of a schema.

The good thing about schemas is that when you access a table that you own (in your own schema), you do not have to refer to the schema name. For instance, you could refer to your table as either one of the following:

EMPLOYEE\_TBL

USER1.EMPLOYEE\_TBL

The first option is preferred because it requires fewer keystrokes. If another user were to query one of your tables, the user would have to specify the schema, as follows:

USER1.EMPLOYEE\_TBL