The SQL() Function

- The SQL function is useful to get data that is not in context, i.e. memory
- It can be used to select data from a table or tables, e.g. "SELECT *field* FROM *table* WHERE ..."

CAUTION:

- **Only use SQL statements to SELECT data from tables**
- **Do not use SQL statements to insert records or update existing records**

1203-V6 – Using Functions and Statements - v02 [41] © LabWare 2011

- A word of caution when using the SQL() function; avoid inserting and updating records in your database using SQL statements.
  - This is not good practice and could lead to data corruption.
  - SQL modifications are **NOT** audited by the system.
- There are numerous **OTHER** LIMS Basic functions that allow you to update data in a more secure way. We will touch on a few of these more secure ways to update a little later on within this course.
- Should you need to update tables with the SQL() function, please note that it is only permitted on user added tables (i.e. User_added = true in Table Master).

Function Demo: SQL()

Purpose: The SQL function executes SQL statements from a LIMS basic routine

Syntax: status = SQL(selectQueryString, "arrayName", maxRecords, cacheFlag, translateConcatinatorOperator, autoLanguagePrefix)

Comments: Similar to what we have seen earlier, arguments that are not needed can be dropped

Example: selectQueryString = "select SAMPLE_NUMBER, TEXT_ID, PRODUCT from SAMPLE where PRODUCT = 'CHOC-CAKE'"
status = SQL(selectQueryString, "sampleArray")

Returns: status = true if the function was successful, false otherwise.
sampleArray contains the returned array values

SQL()

- **selectQueryString** – the SQL statement to be executed
- "**arrayName**" – the array in which the selected records are returned
- **maxRecords** – the maximum number of records returned
- **cacheFlag** – if set to "T", the results are cached in memory (good practice when using SQL queries to populate lists in templates)
- **translateConcatinatorOperator** – if set to "T", LIMS determines the concatinating operator for the current database and replaces the '+' within the SQL, default "T"
- **autoLanguagePrefix** – if set to "T", LIMS returns fields configured with the language prefix replace the specified field, default "F"