Chapter 12 - Macro Functions

“Don’t count the days. Make the days count.”

-Muhammad Ali

The14 rules of Macros

1. Macros can contain one or more SQL Statements

2. Macros are a database Extension in Teradata and not ANSI-Standard

3. Macros do NOT require PERM Space

4. Macros are stored in the Data Dictionary in the USER DBC

5. Every statement in the macro must end in a semi-colon ;

6. Only one DDL Statement can exist in a Macro and it must be the last statement

7. Input Parameters can be passed to the Macro

8. Macros run when users use the EXEC Macro Command

9. Macros can be executed from any viable front end, especially the Nexus

10. Macros can call views and even other Macros

11. All SQL Statements inside a Macro are considered one Transaction

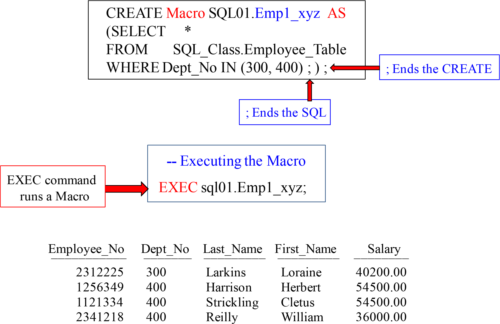
12. Each Macro name must be a unique object within the database it resides

13. USERS need only the EXEC privilege to run a macro

14. No underlying privileges to tables inside the macro need be required

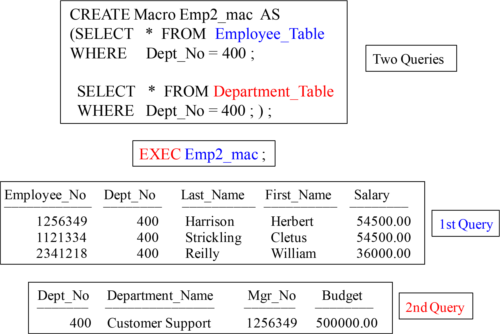
Macros are database extensions and reside in the Data Dictionary in DBC.

CREATINGand EXECUTING a Simple Macro



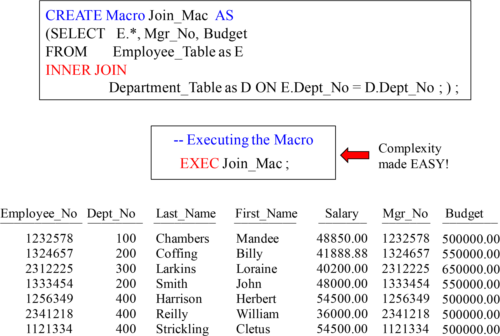
Only ONE transaction will run by executing any macro. Every Macro will have a semi-colon to end each SQL statement and an additional semi-colon to end the CREATE.

MultipleSQL Statements inside a Macro



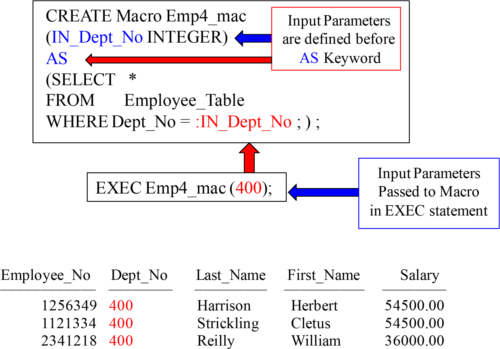
Still, only ONE transaction ran by executing the macro here. Everything in a macro is considered ONE transaction. Notice we have two SQL statements that end with a semi-colon, but we also have a semi-colon to end the CREATE MACRO Statements.

ComplexJoins inside a Macro



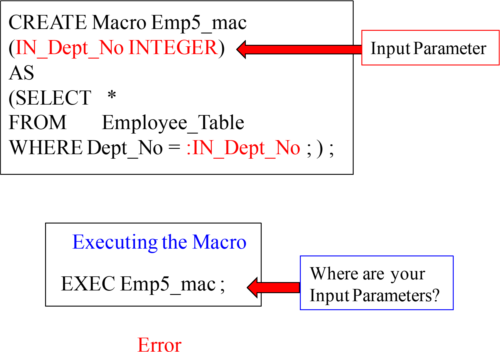
Users can create complex joins in macros and then SHARE them with other Users.

Passingan INPUT Parameter to a Macro



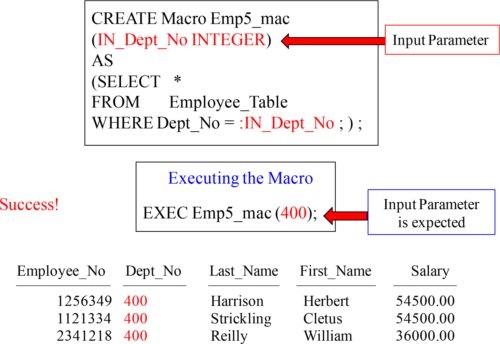
The IN\_Dept\_No represent INPUT Parameters in the above Macro. We must place something within the parentheses or the macro will not work, because it is expecting the USER to tell it the value of IN\_Dept\_No in the EXEC statement.

Troubleshootinga Macro with INPUT Parameters



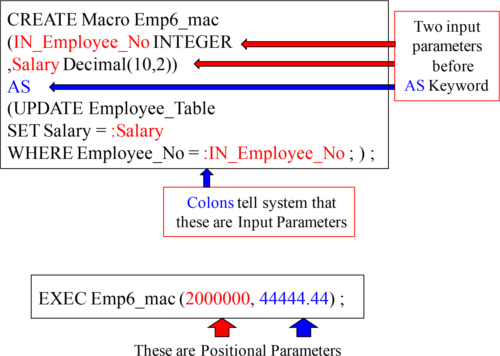
This Macro expected an Input Parameter and errors unless it gets it.

Troubleshootinga Macro with INPUT Parameters



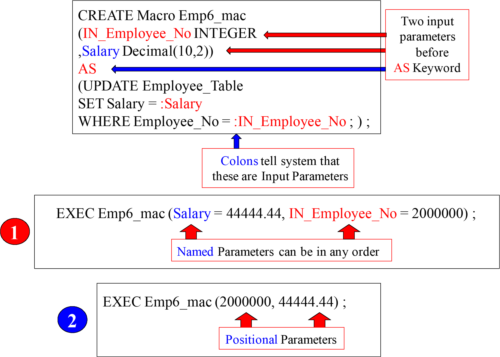
This Macro expected an Input Parameter and errors unless it gets it.

AnUPDATE Macro with Two Input Parameters



This Macro expected two parameters so we placed them in the proper order (Position). Since the CREATE statement listed both, and IN\_Employee\_No was listed first, it will be in the first position for parameters in the EXEC statement inside brackets.

Executinga Macro with Named (Not Positional) Parameters



Both Exec statements are the same except how they pass the Macro Input Parameters.