# Teradata Macro

Macro is a set of SQL statements which are stored and executed by calling the Macro name. The definition of Macros is stored in Data Dictionary. Users only need EXEC privilege to execute the Macro. **Users don't need separate privileges on the database objects used inside the Macro.** Macro statements are executed as a single transaction. If one of the SQL statements in Macro fails, then all the statements are rolled back. Macros can accept parameters. Macro may have numerous SELECT statements and DML's but it can have only one DDL statement and that too must be the single statement.

Macro characteristics:

Can be shared across multiple users and can be secured by providing execute access to required users.

Can be parameterized (arguments can be passed) or non-parameterized.

All statements within macro are processed as one transaction and if one fails that means others will be ROLLBACKED. This is the reason DDL statement must be the single statement inside macro(because DDL cannot be rollbacked).

May improve perfomance time.

Conditions(IF/ELSE) & iterations(FOR/WHILE) are not supported.

Key points to note about Macros:

1. Macros are a Teradata extension to SQL.
2. Macros can only be executed with the EXEC privilege.
3. Macros can provide column level security.

**NOTE: A user must have EXEC privileges to execute the macros. User don't need separate privileges on the macro underling objects like tables, views etc.**

**Difference between Macro and Stored procedure**

| **Macro** | **Stored procedure** |
| --- | --- |
| 1. It will return set of rows to the user | 1. It may return 1 or more values to client as parameters (not rows) |
| 2. May contain BTEQ commands | 2. May contain comprehensive SPL |
| 3. A macro that allows only input values | 3. SP that allows both input and output values |
| 4. It can fetch rows directly | 4. It must use cursors to fetch the rows. |

Create Macros

Macros are created using CREATE MACRO statement.

Syntax

Following is the generic syntax of CREATE MACRO command.

CREATE MACRO <macroname> [(parameter1, parameter2,...)] AS

(

<sql statements>

);

Example

The following example creates a Macro called Get\_Emp. It contains a select statement to retrieve records from employee table.

**Create table emp10(**

**EmployeeNo integer,**

**FirstName varchar(20),**

**LastName varchar(20),**

**Birthdate date);**

Insert into emp10 values(101,'Mike','James','1980-01-10');

Insert into emp10 values(102,'Ram','Singh','1980-01-10');

Insert into emp10 values(103,'Dev','James','1980-01-10');

Insert into emp10 values(104,'July','James','1980-01-10');

Insert into emp10 values(105,'Mark','James','1980-01-10');

select \* from emp10;

\*\*\* Query completed. 5 rows found. 4 columns returned.

\*\*\* Total elapsed time was 1 second.

EmployeeNo FirstName LastName Birthdate

----------- -------------------- -------------------- ---------

103 Dev James 80/01/10

104 July James 80/01/10

101 Mike James 80/01/10

102 Ram Singh 80/01/10

105 Mark James 80/01/10

CREATE MACRO Get\_Emp AS

(

SELECT EmployeeNo, FirstName, LastName

FROM emp10

ORDER BY EmployeeNo;

Select deptno, dname, loc from dept;

);

Executing Macros

Macros are executed using EXEC command.

Syntax

Following is the syntax of EXECUTE MACRO command.

EXEC <macroname>;

Example

The following example executes the Macro named Get\_Emp; When the following command is executed, it retrieves all records from employee table.

EXEC Get\_Emp;

\*\*\* Query completed. 5 rows found. 3 columns returned.

\*\*\* Total elapsed time was 1 second.

EmployeeNo FirstName LastName

----------- ------------------------------ ---------------------------

101 Mike James

102 Robert Williams

103 Peter Paul

104 Alex Stuart

105 Robert James

REPLACE MACRO Get\_Emp AS

(

SELECT EmployeeNo, FirstName, LastName , BirthDate

FROM emp10

ORDER BY EmployeeNo;

);

DROP MACRO Get\_Emp;

Parameterized Macros

Teradata Macros can accept parameters. Within a Macro, these parameters are referenced with **: (colon)**.

Following is an example of a Macro that accepts parameters.

CREATE MACRO Get\_Emp\_Salary (EmployeeNo INTEGER) AS

(

SELECT EmployeeNo, NetPay

FROM Salary

WHERE EmployeeNo = :EmployeeNo;

);

Executing Parameterized Macros

Macros are executed using EXEC command. You need EXEC privilege to execute the Macros.

Syntax

Following is the syntax of EXECUTE MACRO statement.

EXEC <macroname>(value);

Example

The following example executes the Macro named Get\_Emp; It accepts employee no as parameter and extracts records from employee table for that employee.

EXEC Get\_Emp\_Salary(101);

\*\*\* Query completed. One row found. 2 columns returned.

\*\*\* Total elapsed time was 1 second.

EmployeeNo NetPay

----------- ------------

101 36000

REPLACE MACRO Get\_Emp\_Salary (EmployeeNo INTEGER) AS

(

SELECT EmployeeNo, First\_Name, Last\_Name, NetPay

FROM Salary

WHERE EmployeeNo = :EmployeeNo;

);

DROP MACRO Get\_Emp\_Salary;

REPLACE MACRO Weekly\_Report(

p\_empno integer,

p\_dno integer,

p\_deptno integer,

p\_dname varchar(20),

p\_loc varchar(20)) AS

(

select \* from emp1 where empno = :p\_empno;

select \* from dept1 where deptno = :p\_dno;

insert into dept1 values(:p\_deptno, :p\_dname, :p\_loc);

);

DROP MACRO Weekly\_Report;

.set width 200

DROP MACRO dept\_emp\_display;

CREATE MACRO dept\_emp\_display AS

( ECHO '.SET SEPARATOR '' | '' '

;SELECT \* FROM dept;

ECHO '.SET SEPARATOR '' | '' '

;SELECT \* FROM emp;

);

EXEC dept\_emp\_display;

\*\*\* Echo accepted.

\*\*\* Total elapsed time was 1 second.

\*\*\* Query completed. 4 rows found. 3 columns returned.

deptno | dname | loc

----------- -------------- -------------

20 | RESEARCH | DALLAS

30 | SALES | CHICAGO

40 | OPERATIONS | BOSTON

10 | ACCOUNTING | NEW YORK

\*\*\* Echo accepted.

\*\*\* Query completed. 11 rows found. 8 columns returned.

empno | ename | job | mgr | hiredate | sal | comm | deptno

----------- ---------- --------- ----------- -------- ----------- ----------- -----------

7902 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | ? | 20

7567 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | ? | 20

7566 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | ? | 20

7901 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | ? | 20

7788 | SCOTT | ANALYST | 7566 | 87/04/19 | 3000 | ? | 20

7782 | CLARK | MANAGER | 7839 | 81/06/09 | 2450 | ? | 10

7934 | MILLER | CLERK | 7782 | 82/01/23 | 1300 | ? | 10

7839 | KING | PRESIDENT | ? | 81/11/17 | 5000 | ? | 10

7900 | JAMES | CLERK | 7698 | 81/12/03 | 950 | ? | 30

7698 | BLAKE | MANAGER | 7839 | 81/05/01 | 2850 | ? | 30

7369 | SMITH | CLERK | 7902 | 80/12/17 | 800 | ? | 20