**Db2 Stored & SQL Procedures - Episode #6: (In the Db2 User Group – Recap from Robert Catterall’s Tridex)**

**By** [Thomas Halinski](https://community.ibm.com/community/user/people/thomas-halinski) **posted Tue April 08, 2025 02:54 PM**

**Here’s a brief history of Db2 for z/OS Stored Procedures – an initial view of their evolution:**

Db2 for z/OS V4 – The Stored Procedures functionality was introduced.

Db2 for z/OS V5 – The caller of stored procedures can fetch rows from cursor declared and

         opened in the stored procedure.

                               WLM-managed stored procedure address spaces introduced (the

                               Db2-managed stored procedure address space went away with Db2 9 - SPAS).

                               Support for Java Stored Procedures was initiated.

Db2 for z/OS V6 - DDL support was established: CREATE, ALTER, DROP PROCEDURE.

                             The SYSPROCEDURES catalog table replaced by SYSROUTINES.

Db2 for z/OS V7 - SQL Procedure Language (SQL PL) introduced.

Db2 for z/OS V9 -  native SQL procedures

§Key characteristics:

- No external-to-Db2 executable (no object or load module) – a native SQL procedure’s package

is its one and only executable

- Executes in DBM1 (Database Services Address Space – DBAS), the Db2 database services address space (as do all packages) – not in a stored procedure address space

- Runs under caller’s task (an external stored procedure runs under its own TCB in a stored

procedure address space)

- Superior functionality

• Native SQL procedures – not external stored procedures – are where we have seen greatest advances in stored procedure functionality

So now the IBM for z/OS development staff – managers and roll-up-your sleeves developers – decided to really enhance this “feature” of Db2 – let’s create something new and exciting! They created **SQL Procedure Language (SQL PL)** and it was introduced with Version 7. (COBOL programmers were envious, methinks.)

**Can stored procedures be over-used in a Db2 for z/OS environment?**

Over-use is possible – depends on priorities

·       If maximum performance is a priority, stored procedures could be over-used

·       Comparing performance of SQL DML statements issued by client programs versus packaging same statements in server-side stored procedures:

 - Very similar if transaction involves execution of 3-4 SQL DML statements

 - If > 4 SQL DML statements/tran, stored procedures will probably provide better performance

 - If only 1 or 2 SQL DML statements/tran, performance probably best with client-issued SQL

·       Using stored procedures even for transactions that involve execution of only 1 or 2 SQL DML statements could be considered over-use from performance perspective

 - That said, performance difference may not be very substantial

 - If stored procedures will be used even for simplest transactions, use of native SQL procedures becomes more          important (avoid thread task switch overhead of external stored procedures)

 - Note that performance benefit of high-performance DBATs increases as in-Db2 CPU time per transaction decreases.

**What is the difference between functional and stored procedure?**

In a function, it is mandatory to use the RETURNS and RETURN arguments, whereas in a stored procedure is not necessary. In few words, a stored procedure is more flexible to write any code that you want, while functions have a rigid structure and functionality.Feb 14, 2017

What are the advantages of functions over stored procedure?

The only advantage of function is we can use function as inline queries.  
...

**Advantages of User Defined Functions over Stored Procedures**

* stored procedures are more advantageous than functions.
* functions have limited error handling.
* functions cannot use temporary tables.
* functions cannot call stored procedures.

A screenshot of a computer program

AI-generated content may be incorrect.

[#IBMChampion](https://community.ibm.com/community/user/search?s=tags%3A%22IBMChampion%22&executesearch=true)