Stored Procedures (proc)

A stored procedure is a named (for repeated usage) PL/SQL block which performs one or more specific task.

Stored procedure is set of SQL code which is stored in database server and can be invoked by program or trigger or stored procedure itself.

in other words stored procedure is way to execute your business logic directly into database server.

This is similar to a procedure in other programming languages.

A procedure has a header and a body.

* The header consists of
  + The name of the procedure and
  + The parameters or variables passed to the procedure.
* The body consists of (similar to a general PL/SQL Block)
  + declaration section,
  + execution section and
  + exception section

We can pass parameters to procedures in three ways.

* IN-parameters
* OUT-parameters
* IN OUT-parameters

A procedure may or may not return any value.

General Syntax to create a procedure is:

CREATE [OR REPLACE] PROCEDURE proc\_name [list of parameters]

IS

Declaration section

BEGIN

Execution section

EXCEPTION

Exception section

END;

The syntax within the brackets [ ] indicate they are optional.

IS - marks the beginning of the body of the procedure and is similar to DECLARE in anonymous PL/SQL Blocks.

Procedures: Example

CREATE PROCEDURE procedure1 /\* name \*/

(IN parameter1 INTEGER) /\* parameters \*/

BEGIN /\* start of block \*/

DECLARE variable1 CHAR(10); /\* variables \*/

IF parameter1 = 17 THEN /\* start of IF \*/

SET variable1 = 'birds'; /\* assignment \*/

ELSE

SET variable1 = 'beasts'; /\* assignment \*/

END IF; /\* end of IF \*/

INSERT INTO table1 VALUES (variable1);/\* statement \*/

END /\* end of block \*/

To execute a Stored Procedure?

There are two ways to execute a procedure.

* From the SQL prompt.
  + EXECUTE [or EXEC] procedure\_name;
* Within another procedure – simply use the procedure name.
  + procedure\_name;

Trigger:

<https://www.codeproject.com/Tips/624566/Differences-between-a-Stored-Procedure-and-a-Trigg>

We can execute a stored procedure whenever we want with the help of the exec command, but a trigger can only be executed whenever an event (insert, delete, and update) is fired on the table on which the trigger is defined.

We can call a stored procedure from inside another stored procedure but we can't directly call another trigger within a trigger. We can only achieve nesting of triggers in which the action (insert, delete, and update) defined within a trigger can initiate execution of another trigger defined on the same table or a different table.

Stored procedures can be scheduled through a job to execute on a predefined time, but we can't schedule a trigger.

Stored procedure can take input parameters, but we can't pass parameters as input to a trigger.

Stored procedures can return values but a trigger cannot return a value.

We can use Print commands inside a stored procedure for debugging purposes but we can't use print commands inside a trigger.

We can use transaction statements like begin transaction, commit transaction, and rollback inside a stored procedure but we can't use transaction statements inside a trigger.

We can call a stored procedure from the front end (.asp files, .aspx files, .ascx files, etc.) but we can't call a trigger from these files.

Stored procedures are used for performing tasks. Stored procedures are normally used for performing user specified tasks. They can have parameters and return multiple results sets.

The Triggers for auditing work: Triggers normally are used for auditing work. They can be used to trace the activities of table events.