The Difference between stored procedure and :

A stored procedure in SQL Server is a collection SQL statements and sql command logic, which is compiled and stored on the database server. The use of stored procedures to implement SQL queries direct on the server itself rather than in the code and remove use of embedded SQL queries. Stored procedured can also improve performance of an application.

SQL CREATE PROCEDURE command is used to create a stored procedure. Here is the syntax of a stored procedure:

1. **CREATE** **PROCEDURE** <Procedure\_Name, sysname, ProcedureName>
2. -- Add the parameters for the stored procedure here
3. <@Param1, sysname, @p1> <Datatype\_For\_Param1, , **int**> = <Default\_Value\_For\_Param1, , 0>,
4. <@Param2, sysname, @p2> <Datatype\_For\_Param2, , **int**> = <Default\_Value\_For\_Param2, , 0>
5. **AS**
6. **BEGIN**
7. -- SET NOCOUNT ON added to prevent extra result sets from
8. -- interfering with SELECT statements.
9. **SET** NOCOUNT **ON**;
10. -- Insert statements for procedure here
11. **SELECT** <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
12. **END**

If you're new to stored procedures, learn here [Create and Execute Stored Procedures In SQL Server](https://www.c-sharpcorner.com/article/how-to-create-a-stored-procedure-in-sql-server-management-studio/).

A SQL triggers are database objects similar to stored procedures. The difference is, triggers are set and fired (executed) on specific events such as when a record is inserted into a table, a trigger can be fired and the SQL query written in the trigger will be executed.

The CREATE TRIGGER SQL command is used to create a new trigger on a table.

1. **create** **trigger** saftey
2. **on** **database**
3. **for**
4. create\_table,alter\_table,drop\_table
5. **as**
6. print'you can not create ,drop and alter table in this database'
7. **rollback**;

Here are some of the key differences between SQL stored procedures and triggers.

1. When you create a trigger, you must identify an event and action of your trigger when the trigger will be executed. Stored procedures are just SQL statements that don't require an event and action. Stored procedures can be called direct from an application or within other SQL commands including other stored procedures.
2. A trigger runs automatically when the event is fired. A stored procedure is executed manually or from a caller application.
3. Within a trigger you can call stored procedures but you cannot call a trigger from a stored procedure.
4. A trigger executes implicitly whereas a stored procedure is executed via a procedure call from another block.
5. We can call a stored procedure from front end (.asp files, .aspx files, .ascx files etc.) but we can't call a trigger.
6. A stored procedure can take the input parameters, but we can't pass input parameters to a trgger.

A SQL trigger is a database object which fires when an event occurs in a database. We can execute a SQL query that will "do something" in a database when a change occurs on a database table such as a record is inserted or updated or deleted. For example, a trigger can be set on a record insert in a database table. For example, if you want to increase the count of blogs in the Reports table when a new record is inserted in the Blogs table, we can create a trigger on the Blogs' table on INSERT and update the Reports table by increasing blog count to 1.

Types of Triggers

There are two types of triggers:

1. DDL Trigger
2. DML Trigger

DDL Triggers

The DDL triggers are fired in response to DDL (Data Definition Language) command events that start with Create, Alter and Drop, such as Create\_table, Create\_view, drop\_table, Drop\_view and Alter\_table.

**Code of a DDL Trigger**

create trigger saftey

on database

for

create\_table,alter\_table,drop\_table

as

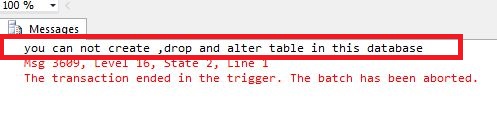
print'you can not create ,drop and alter table in this database'

rollback;

SQL

Copy

When we create, alter or drop any table in a database then the following message appears:



DML Triggers

The DML triggers are fired in response to DML (Data Manipulation Language) command events that start with Insert, Update, and Delete. Like insert\_table, Update\_view and Delete\_table.

create trigger deep

on emp

for

insert,update,delete

as

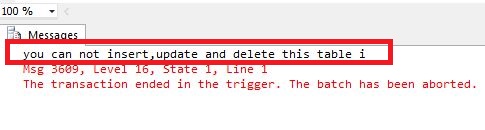
print'you can not insert,update and delete this table i'

rollback;

SQL

Copy

When we insert, update or delete in a table in a database then the following message appears,



There are two types of DML triggers

AFTER Triggers

AFTER triggers are executed after the action of an INSERT, UPDATE, or DELETE statement.

create trigger insertt

on emp

after insert

as

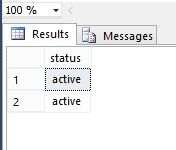
begin

insert into empstatus values('active')

end

SQL

Copy



INSTEAD Of Triggers

It will tell the database engine to execute the trigger instead of executing the statement. For example an insert trigger executes when an event occurs instead of the statement that would insert the values in the table .

CREATE TRIGGER instoftr

ON v11

INSTEAD OF INSERT

AS

BEGIN

INSERT INTO emp

SELECT I.id, I.names

FROM INSERTED I

INSERT INTO emp1values

SELECT I.id1, I.name1

FROM INSERTED I

END

SQL

Copy

When we insert data into a view by the following query then it inserts values in both tables :

insert into v11 values(1,'d','dd')

SQL

Copy

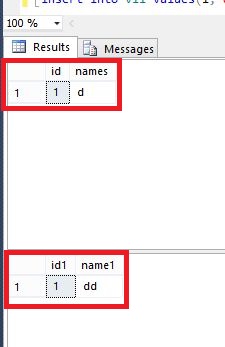
You can see both tables by the folowing query:

select \* from emp

select \* from emp1values

SQL

Copy



# **Diference Between Stored Procedures And Functions :**

Introduction

Both stored procedure and functions, are database objects that contain a set of SQL statements to complete a specific task. But there are also some differences among them. We will discuss their differences and their usage in this article.

Stored Procedures

Stored procedures are pre-compiled objects which are compiled for the first time and its compiled format is saved. It will execute whenever it is called. If you think of a query that you write over and over again, instead of writing that query each time you save your query as a stored procedure and whenever you want that query to be executed you just call saved store procedure. You can also pass parameters to the stored procedure. So, it becomes more flexible and you can do what your parameter say in your stored procedure.

Stored Procedures can also improve performance. Many tasks are implemented as a series of SQL statements. The logic applied to result of first SQL statements determine which subsequent SQL statements are to be executed. If these SQL statements and conditional logic are written into a Stored procedure, they become part of a single execution plan on the server. All of your work will be done on the Server which makes it faster.

Advantages of Stored Procedures

**Performance**

Stored procedures are compiled once and stored in executable form, so procedure calls are quick and efficient. Executable code is automatically cached and shared among users. This lowers memory requirements and makes application fast.

**Productivity and Ease of Use**

By passing parameters to stored procedures, you can handle your logic in stored procedures. You can use the IDE of your choice and then you can deploy them on any tier of network architecture. Moreover, they can be called by programmatic interfaces and development tools.

**Security Controls**

You can grant user permissions to execute a Stored Procedure independently of underlying table permissions.

Functions

Like functions in programming languages, SQL Server user-defined functions are routines that accept parameters, perform an action such as a complex calculation, and return the result of that action as a value. The return value can either be a single scalar value or a result set. A function is compiled and executed every time whenever it is called. A function must return a value and cannot modify the data received as parameters.

There are two types of Functions,

* *Built-in functions*Operate as defined in the Transact-SQL Reference and cannot be modified. The functions can be referenced only in Transact-SQL statements using the syntax defined in the Transact-SQL Reference.
* *User-Defined Functions*By using the CREATE FUNCTION statement, users can write their own Transact-SQL functions according to their specific need. User-defined need zero or more input parameters and return a single data value.

Advantages of Functions

**Modular Programming**

You can create a function once, store it in your database and use it whenever you need.

**Faster Execution**

Similar to Stored Procedures, Transact-SQL User-Defined Functions reduce the compilation cost of Transact-SQL code by caching the plans and reusing them for repeated executions. This makes it faster.

Difference between Store procedure and functions

1. The function must return a value, but in Stored procedure it is optional. Even a stored procedure can return zero or n values.
2. Functions can be called from Stored procedures while a Stored procedure cannot be called from a function.
3. The procedure allows to write INSERT, UPDATE, DELETE statements with SELECT statement while function only allows SELECT statement.
4. Procedures cannot be utilized in a SELECT statement while a function can be embedded in a SELECT statement.
5. Stored procedures cannot be used in SQL statements like WHERE/HAVING/SELECT statement whereas functions can be used.
6. We can use transactions in Stored procedures while transactions cannot be used in functions.

Code Examples

Printing Hello World in stored procedure vs. in function.

In the stored procedure,

1. **CREATE** **PROCEDURE** HelloWorldprocedure
2. **AS**
3. PRINT 'Hello World'

Executing stored procedure

1. **exec** HelloWorldprocedure

In function,

1. **CREATE** **FUNCTION** dbo.helloworldfunction()
2. **RETURNS** **varchar**(20)
3. **AS**
4. **BEGIN**
5. **RETURN** 'Hello world'
6. **END**

Using functions,

1. **select** dbo.helloworldfunction() **as** regards

Using variable in Stored Priocedure vs. in Function

In the stored procedure,

1. **CREATE** **PROCEDURE** CONVERTCELSIUSTOFAHRENHEIT
2. @celsius **real**
3. **as**
4. **select** @celsius\*1.8+32 **as** Fahrenheit

Executing a stored procedure,

1. **exec** CONVERTCELSIUSTOFAHRENHEIT 0

In function,

1. **CREATE** **FUNCTION** dbo.f\_celsiustofahrenheit(@celcius **real**)
2. **RETURNS** **real**
3. **AS**
4. **BEGIN**
5. **RETURN**  @celcius\*1.8+32
6. **END**

Using function,

1. **select** dbo.f\_celsiustofahrenheit(0) **as** fahrenheit

In this article, we discussed what stored procedures and functions are, what are their differences, and how they are used.

**3- drop and delete statement :**

Difference between DELETE and DROP SQL. DELETE is a Data Manipulation Language command, DML command and is used to remove tuples/records from a relation/table. Whereas DROP is a Data Definition Language, DDL command and is used to remove named elements of schema like relations/table, constraints or entire schema.

**4- select and select into statement**