

Title

SQL and Database Design for web development - Chapter IV.

About the Author

Sérgio Fontes it's a Software & Database Analyst engineer. With academic background in Maths and Informatics, Sérgio worked on several projects under the EC's IST (Information Society Technologies) programme as a programmer and also as project manager of the Portuguese team.

Abstract

Usually referred as global variables the @@variables are used to return different kinds of information about what's going on in SQL Server. The use Control of flow syntax like, if..else, begin..end or while..., is used to branch and loop as necessary, let see how we can use it on our projects.

Keywords

SQL; T-SQL; Database; ER; Redundancy; Normalization; Websites; Data; Web; User-defined functions; variables:

About this Article

In this series of articles I want to analyze and describe database and query development for web solutions, as development technology we will use SQL and Microsoft SQL Server 2000 as the database engine. After making a small trip on the SQL basics we are read to take a look on T-SQL language. On this article we will work with T-SQL control structures and Global variables, and see how we can profit by using them on our development.

Content

The number of rows affect by a certain T-SQL statement it's a very information detail for you code or only for you programming task and not for the final code version. If you want to fill a table or a list you will probably need to know the number of rows affected. To retrieve that information you can use the @@ROWCOUNT function. Maybe an example can help you: Let us make a query to the clients table created on the Chapter III. Without the variable we would use:

```
SELECT * FROM clients
```

The result would be:

	client_id	client_name	client_phone
1	1	Client A	+351989894876
2	2	Client B	+351234567678

And we would have no knowledge about the number of rows that this query result contains, unless you use the necessary code on VB,C# or JAVA. Let's try a different statement:

```
SELECT * FROM clients
SELECT @@ROWCOUNT AS NumInCA
```

And the result would be:

	client_id	client_name	client_phone
1	1	Client A	+351989894876
2	2	Client B	+351234567678

	NumInCA
1	2

As i mentioned, this additional information can be very helpful. Let's take a look at a different global variable that I use a lot and on web development will allow you to reduce some lines of code.

The @@IDENTITY allows you to retrieve that id of the record that you are adding to the database. Many times on web development, we need to insert a new record and in the same block of code we need to use the id of this new record. This variable helps you to retrieve that information using one single T-SQL statement and without the need to use additional code of any other programming language. On the following example we will create a new record on the table products created on Chapter I, but we will make a small change on its structure(the primary key will be an increment number) and retrieve its new id.

```
INSERT INTO products (prod_name,prod_quality) values ('product B','bad')
SELECT @@IDENTITY AS NewProductID
```

The result of this statement is:

	NewProductID
1	3

If you make a select statement to the products table defining the following where clause, produc_id=3, you can check that it matches the inserted item.

Now let's take a look at the control structures, also called as control flow. Lets start with a simple one, the IF..ELSE control and use variables to simulate data. Suppose that we make a query to a client account and we want to publish if this client is a GOOD or BAD client, the rule for this client to be classified as GOOD is that the value of product purchase is higher then 500000. Let's simulate that this client as a purchase value of 450000. How can we classify him using T-SQL?

```
DECLARE @purchasevalue integer
SET @purchasevalue = 450000
IF @purchasevalue>500000
SELECT 'GOOD' AS STATUS
ELSE
SELECT 'BAD' AS STATUS
```

This is a simple example on how to use the IF..ELSE control structure and I am certain that you will find opportunities to use it in a more complex way on your projects. In this statement you can use nested statements or any other kind of variables and functions previously discussed.

A very powerful is the WHILE control statement, I like to use it every time I need to populate a database for development testing. In this next example we will use the WHILE syntax to insert records on the products table to achieve 1000 records.

```
WHILE (SELECT COUNT(*) FROM products)<1000
BEGIN
    INSERT INTO products (prod_name,prod_quality) VALUES ('Product C','Bad')
    if (SELECT COUNT(*) FROM products)>1000
        BREAK
    ELSE
        CONTINUE
END
```

Well this last example as something more than only the WHILE statement but shows the power of T-SQL. We can see a combination of functions, parameters and control structures for one single goal. This type of complexity is mainly used on triggers and stored procedures that we will discuss on the chapter IV, the last one of this set of 5 articles. On the next article we will discuss the benefit of using Triggers and Stored Procedures.

Conclusion

T-SQL is without question a very powerful language, the use of global variables allows you to have full knowledge of what is happening on the database server. Control structures like IF..ELSE and WHILE provide to the developer a dynamic feature to execute multiple instructions with one simple call to the sever reducing the number of client-server request. On Web development this latest reference must be taking under hard consideration because web servers have a limit number of connections.

Contact Info

Sérgio Fontes
Sergio@computencial.com