

Title

SQL and Database Design for web development - Chapter III.

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

The use of T-SQL variables, control structures and user defined functions permits to execute a set of instructions on the server side. This option reduces the number of instructions on the client side and also the size of packets to be transported trough the internet between client and server side. This means that less information needs to be used on the client side accelerating the process of publishing the data.

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 ready to take a look on T-SQL language. On this article we will work with T-SQL variables, control structures, parameters and functions and see how we can profit by using them on our development.

Content

As mentioned on chapter II T-SQL is similar to common programming languages like VB, VB .NET, Java, C# or C++. Like them we can work with variables, control structures, parameters and functions. Let's take a small look at these interesting and powerful features.

To do this lets create a new table on our database using the following T-SQL statement

```
CREATE TABLE [dbo].[Clients] (  
    [client_id] [int] NOT NULL,  
    [client_name] [char] (100) COLLATE Latin1_General_CI_AS NULL,  
    [client_phone] [char] (100) COLLATE Latin1_General_CI_AS NULL  
) ON [PRIMARY]
```

And insert some records,

```
INSERT INTO clients (client_id,client_name,client_phone) values (1,'Client A','+351989894876')  
INSERT INTO clients (client_id,client_name,client_phone) values (2,'Client B','+351234567678')
```

Now let's suppose that we need to add some description to the data that we want to publish, for example indicate the type of phone. To do so, we need the following statement:

```
DECLARE @local varchar(12)
SET @local = 'Client Phone: '
SELECT Client_Name, @local + Client_Phone AS Phone
FROM Clients
```

Well I agree with you that we need some imagination to think that this is a usable statement on real code. Let's try a more elaborated solution:

Imagine that you are developing a multi language website. And that the page identifying labels are published at the same time as the dynamic data. You can use one single statement instead of 2 statements. To do that you just need to replace:

```
SET @local = 'Client Phone: '
```

By

```
SET @local = (SELECT phonelabel FROM labeltranslation WHERE lng='ENG')
```

Assuming that exists the table languagetranslation.

The use of variables combined with the use of T-SQL functions it's a very powerful and allow you to make operations on the server side that are time consuming and can represent large amount of data to be transferred between Client and Server Side.

Another sample of work that can be done directly on the server side, preventing you from the need to do it on your application code, after the query statements is data conversion. Let's take a look at some samples:

```
DECLARE @price float
SET @price = 345.56765456
SELECT @price,
ROUND(@price, 0) AS Rounded,
ROUND(@price, 0, 1) AS Truncated,
ROUND(@price, 1) AS Approximate,
ROUND(@price, -1) AS Approximate2
```

Where is the result of this T-SQL statement:

price	Rounded	Truncated	Approximate	Approximate2
345.56765455999999	346.0	345.0	345.60000000000002	350.0

The above example uses the *ROUND* function but you have other functions that allow you to format String, Date and Numeric values. Return to our invoice header and invoice details tables mentioned on Chapter I, we can find a lot of use for T-SQL functions. For example to determine the total value of each item

on the invoice detail and also to format, the final result according with the client side regional settings.

Date and time information is something that we are always using on our applications. Its good practice that every time you make an insert statement on you database to register the date and time as well as the user id. If you chose not to use T-SQL you first need to format your data and then add it to you SQL statement. If you choose to use T-SQL, that I really advise, you just need to use *GETDATE()* function. Let's see how it works:

```
SELECT
DATEPART(dy, GETDATE()) AS DayOfYear,
DATEPART(dd, GETDATE()) AS DayNum,
DATEPART(ww, GETDATE()) AS WeekNum,
DATEPART(dw, GETDATE()) AS Weekday,
DATEPART(hh, GETDATE()) AS Hour,
DATEPART(mi, GETDATE()) AS Minute,
DATEPART(ss, GETDATE()) AS Seconds
```

And the result is:

	DayOfYear	DayNum	WeekNum	Weekday	Hour	Minute	Seconds
1	120	30	18	7	17	5	13

Now you just need to use it according to your needs, and reduce the number of instructions and data to be transferred from client side to server side. You will find that you requests to the database will much faster.

Also common is the need to make operations between date and time values, in programs that happen a lot. I must say that in the beginning I had some hard time but now I simple use the following:

```
DECLARE @invoicepaydate datetime
DECLARE @today datetime
SET @today =getdate()
SET @invoicepaydate = '2005-01-01'
SELECT DATEDIFF(dd,@invoicepaydate,@today) AS PayDelayedDays
```

And this is the result considering that today is 2005-04-30

	PayDelayedDays
1	119

Well we could see much more examples but I believe that the samples I showed in this article give a general idea of what we can do using T-SQL. On the next chapter I will present some samples on how to work with control structures like *IF..ELSE*, *GOTO* and some functions on how to retrieve information about our statement status.

Conclusion

The use of T-SQL variables, functions and parameters allows you to improve your code and server response. As mentioned on Chapter II and in this article, T-SQL as common features to other programming languages but remember it

doesn't replace their use, but can help you to develop code that as a better performance.

Contact Info

Sérgio Fontes
Sergio@computencial.com