

Title

SQL and Database Design for web development - Chapter V.

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

Triggers and Stored Procedures are powerful tools to manage changes on the database and also, to increase the database performance. Another powerful strategy is the use of Indexes to increase the performance on database requests. However this is a complex feature to develop and even understand.

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. A long way trough SQL and T-SQL was made during these 5 articles. To finish in high standards we need to talk about Triggers, Stores Procedures and Indexes. It's a very important issue if you want deliver state of the art database and also state of the art software that uses database servers as data repository. Due to the complexity of these features I will focus on its properties, purposes and present some simple examples.

Content

The importance of Triggers

The use of Triggers is what I like to call, the creation of Active Databases. This "strange" feature allows you to create mechanisms to deal with changes on the database. Triggers are mainly used to check data integrity, this means, that a trigger can allow or cancel an SQL statement. But triggers can be used on a different purpose like the following:

Suppose that you want to check the invoices status of a certain client when this client is making a new purchase. If you use a trigger that runs every time a new invoice is created, you can verify the client account and determines if the engine can or cannot create the new invoice. For example using a IF..ELSE control to compare the amount with a user defined value.

The use of Stored Procedures

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. They are used to encapsulate a set of operations or queries to execute on a database server. This means that you can put together a set of operations inside the same piece of code. The use of stored procedures can increase the performance of a database because SQL server compiles the stored procedure once and then reuses its execution plan.

Here is an example of a simple stored procedure:

```
CREATE PROCEDURE GetInventory
@location varchar(10)
AS
SELECT Product, Quantity
FROM products
WHERE Warehouse = @location
```

The variable @location receives an outside values that is passed every time the stored procedure is called. If you need to pass more values to the stored procedure you just need to create more variables inside the block of code.

Here is a small sample on how to call a stored procedure from an ASP page:

```
<%
Set cn = Server.CreateObject("ADODB.Connection")
Set cmd = Server.CreateObject("ADODB.Command")
cn.Open "data source name", "userid", "password"
Set cmd.ActiveConnection = cn
cmd.CommandText = "sp_test"
cmd.CommandType = adCmdStoredProc
cmd.Parameters.Refresh
cmd.Parameters(1) = 11
cmd.Execute
%>
```

Hope this helps on your future developments.

Optimize your database using Indexes

The use of this features allow you to increase the performance of you database. Every time you create a key on a table object you are creating automatically an index on that table. In this article we will discuss Hash-Based Indexing and Tree-Based Indexing.

The use of Hash-Based Indexes allows you to search a tables according with a search key value. For example if we create an Index on the table Clients we can retrieve all records called Consulting.

Tree-Based Index is an alternative to the hash index, this index organizes the records on a Tree data structure then a hierarchical search is taken.

Conclusion

It's been long way since we started on Chapter I, my goal was to give a clear view of Database Development and the uses of SQL and T-SQL languages. My approach by using the web development to explain certain choices was in my opinion a good one. It's not easy to explain to database designers that they must achieve full normalization and then say that in certain cases we must live with data redundancy. On Chapter II, a little static in my opinion one was my intention to give a small recycling on simple SQL statements. It was imperative before our small diving on the T-SQL language and some of its features.

I was concerned about the best way to present all this information and above all, after all i wanted to make a real connection between theory and the real world. In school or training activities we receive mass injections on how to do it but the small details on how to react on particular situations are many times forgotten.

The use of SQL and T-SQL can be a big advantage for you future developments. Software design and implementation is not all about deliver the project no matter the product performance. And it's our job as developers to be rational and if necessary spend more time on development, on a long term basis it's the better choice. In future articles I will dedicate to this latest chapter in more details, because there is a lot to say and we will need more that a few set of pages.

Contact Info

Sérgio Fontes
Sergio@computencial.com