

Title

SQL and Database Design for web development - Chapter II.

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

Developed by IBM, Structured Query Language is a very powerful tool to manage data. You can perform a set of operations that will allow you to create, alter or delete tables and also insert, update and delete records. Database Vendors like Microsoft and Oracle have developed implementations for their own database products like T-SQL and PL-SQL.

Keywords

SQL; T-SQL; Database; ER; Redundancy; Normalization; Websites; Data; Web;

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. On the first article we discuss certain aspects of database design and also mentioned some SQL statements. In this article I will present the SQL language and T-SQL as well as set of SQL statements (DML and DDL subsection) so that you can work with your databases. My objective is to take a small trip on SQL statements so that in Chapter III we can look in more detail to certain aspects of T-SQL.

Content

SQL what is this? And T-SQL?

Structured Query Language (SQL) is the most widely used commercial relational database language. It was developed by IBM and the current ANSI/ISO standard is called SQL: 1999.

SQL is divided on 8 subsets:

1. DML – Data Manipulation Language where you can find the *insert*, *delete* and *update* statements for rows (table items).
2. DDL – Data definition language where you can find *create*, *delete* and *update* statements for tables and views.
3. Triggers and Advanced Integrity Constraints where you can create actions that will be triggered whenever the database changes.

4. Embedded and Dynamic SQL allowing you to call SQL code from a host language like C and COBOL.
5. Client-Server Execution and Remote Database Access where you can use commands that control how a client application program can connect to an SQL Server.
6. Transaction Management.
7. Security where you can control user access to data objects like tables and views.
8. Advanced features like data mining

As you can see with SQL you have full control over any database, and with T-SQL?

Transact-SQL is the SQL Server implementation of SQL-92, a standard codified by the American National Standards Institute (ANSI) and also adopted by the International Organization for Standardization (ISO). It's similar to a programming language, as no user interface but as the advantage of running on the server, this means that improves performance because less data has to traverse the network for processing on the client.

Probably you have also seen mentions to PL-SQL which is the "same" as T-SQL but as an implementation of Oracle Database Server.

Now that we have some notions of what is SQL and T-SQL let's see how does it look like. Using the Tables mentioned on Chapter I let's see how can we create these tables and manipulate records.

SQL Statement to create the Products table:

```
CREATE TABLE quality      (      quality_name      CHAR(10)
                                PRIMARY KEY          (quality_name)
                                )

CREATE TABLE products     (      prod_id           INTEGER,
                                Prod_name           CHAR(10),
                                prod_quality        CHAR(10)
                                PRIMARY KEY          (prod_id),
                                FOREIGN KEY          (prod_quality) REFERENCES quality
                                )
```

Once we have the tables created then we can insert records using the following statements:

SQL insert Statement:

```
INSERT INTO quality (quality_name) values ('excellent')
INSERT INTO quality (quality_name) values ('good')
INSERT INTO quality (quality_name) values ('normal')
INSERT INTO quality (quality_name) values ('bad')
INSERT INTO products (prod_id,prod_name,prod_quality) values (1,'product A','bad')
```

In this case the insert statement on the table products runs successfully because the prod_quality value that we want to add exists on the table quality, but if we try to insert a value that doesn't exist on that table the following message will be returned:

Violation of PRIMARY KEY constraint 'PK_products'. Cannot insert duplicate key in object 'products'.

To display the records that we have already inserted you need to use the select statement:

```
SELECT * FROM PRODUCTS
```

Or

```
SELECT prod_id,prod_name,prod_quality FROM PRODUCTS
```

If want a specific record, for example if you want to see all the bad quality products, you just need to add the where clause as in the following example:

```
SELECT * FROM PRODUCTS WHERE prod_quality='bad'
```

If we want to update a record then we need to use the following statement:

```
UPDATE products SET prod_name='Product B' WHERE prod_id=1
```

This is a very dangerous statement, many times developers forget to use the where clause and then they simple update all the records on the table. This common error happens also when we want to delete a certain record:

```
DELETE products
```

Instead of

```
DELETE products WHERE prod_id=1
```

The insert, update and delete statements are the basic SQL statements on the DML section but more elaborated statements can be done like the inner join statement and the nested statement, let's see some examples:

The inner join statement is used to query multiple tables that have related records, here is one example:

```
SELECT * FROM products INNER JOIN quality ON products.prod_quality=quality._quality_name
```

The nested statement is a statement that as another statement embedded within it, the embedded query is also called as subquery.

```
SELECT prod_id,prod_name, (SELECT quality_name WHERE quality_name=products.prod_quality) AS  
prodquality FROM products
```

Conclusion

SQL it's a very simple language that allows you to operate with tables, records and order database objects. It's your choice to use this code on your third party application or directly on the database server.

Contact Info

Sérgio Fontes

Sergio@computencial.com