THE HANDLING OF DATA STORED IN DATABASES

Lecturer Marius-Cristian APETREI PhD. candidate
“Dimitrie Cantemir” Christian University
Faculty of Foreign Languages, Bucharest
Email: marius.apetrei@gmail.com

Lecturer Bogdănel Marian DRĂGUȚ PhD
“Dimitrie Cantemir” Christian University,
Faculty of Tourism and Commercial Management, Bucharest
Email: dragut_bogdan@yahoo.com

Dominic PEREZ-DANIELESCU, University tutor
Dimitrie Cantemir” Christian University,
Faculty of Tourism and Commercial Management, Bucharest
Email: domynick2005@yahoo.com

Abstract:
This paper aims at describing some techniques for manipulating data stored in databases. At first, it is important to describe the main features of a network, their classification, the client-server concept, and the data manipulation language. The paper is divided into two parts: the first part deals with the ADO.NET object, and its characteristics, and the second part deals with seeking the methods for handling data stored in the database.

Key words: client-server concept, network, component unit, language, model, connection.

JEL Classification: C88

At first, the computer network term meant connecting multiple terminals with a powerful computer. Usually, there was a lack of computing power terminals, all operations being carried out by the central computer. The main characteristics of a computer network are the type, location, and the access way to side resources. In terms of components location the computer network can be classified as follows:

The client-server concept includes the existence of at least two manufacturing processes running in various run-time environments and communicate with each other in one respect: one of the processes (the client) sends a processing request to the other process (the server), sending back the resulting processing afterwards.

Applications are classified by:
1. Application layers (can be 2, 3 or n layers);

In a database application, the two processes are:
1. The client - an application written in a programming language, using the data stored in the database;
2. The server - used for data storage.

By the location components, computer networks can be classified as follows:

- Local Area Networks (LAN - Local Area Network) – the distances between stations are of order kilometers at most;
- metropolitan networks (MAN - Metropolitan Area Network) - the distances between stations can reach tens of kilometers;
- large geographic area networks (WAN - Wide Area Network) - the distances can be thousands of kilometers; the area covered by this type of networks may correspond to a country or even to the whole globe.

The two processes are connected through two components:

1. The network component - provides the fundamental aspects of communication between two processes on the computer;
2. API for inter-connection - is a set of software components that implement an API (Application Programming Interface) which provides data transfer between the applications and the client-server process.

The data manipulation language (DML) provides a set of methods that allow basic operations to manipulate database data:

- inserting new data,
- data changes,
- retrieving data,
- deleting data.

The DML languages can be of two types: procedural and non-procedural.

- The procedural ones show how the result of DML instructions can be obtained;
- The non-procedural ones describe only the results to be obtained.

1. ADO.NET object model

ADO.NET (ActiveX Data Objects) is a component of the nucleus. NET Framework that allows to connect to different data sources, retrieval, manipulation and updating.

Usually, the data source is a database, but it could also be a text file, an Excel spreadsheet, an Access file or an XML file.

ADO.NET allows both connected work style and disconnected work style, the applications being connected to the server database only through retrieving and updating data. This allows reducing the number of simultaneous open connections to the data sources.

ADO.NET provides the tools of XML use and representation to transfer data between applications and data sources, providing a common data representation, allowing access data from different sources of different types and their processing as entities, without the need to explicitly convert data in XML format or vice versa.

These characteristics are determined in settling the benefits provided by the ADO.NET:

- Interoperability. ADO.NET can easily interact with any component that supports XML.
- Durability. ADO.NET enables an application development architecture due to the way of data transfer between architectural levels.
- Programability. ADO.NET simplifies programming for different tasks such as SQL,
which leads to an increase in productivity and a decrease in the number of errors.

- Performance. The explicit conversion is no longer necessary to transfer data between applications, which leads to increasing performance.
- Accessibility use disconnected architecture enables simultaneous access to the same data set. Reducing the number of open connections simultaneously determine the optimal use of resources.

Server NET. NET Framework Data Provider for SQL includes interaction with Microsoft SQL Server, Oracle Data Provider for Oracle databases and OLE DB Data Provider for accessing databases that uses OLE DB technology for displaying data (e.g. Access, Excel or SQL Server version earlier than 7.0). Data provider allows an application to connect to the data source, executes commands and saves the results. Each data provider includes Connection, Command, DataReader and DataAdapter components.

Before any operation with an external data source, you must make a connection (link) to the source. Class Connection classes (SqlConnection, OleDbConnection etc.) contain data on the data source (location, access, your account name and password etc.), methods for opening/closing the connection, starting a transaction etc. These classes are the subspaces (SqlClient, OLEDB etc.) of the space System Data space.

In addition, it implements the IDbConnection interface.

To open a connection one can programmatically instantitate an object of type connection, and a parameter specifying a string containing information about the connection.

Examples of connecting to a SQL data source: using System.Data.SqlClient;
SqlConnection co = new SqlConnection();
co.ConnectionString = “Data Source=localhost; User ID=profesor;pwd=info; Initial Catalog=Orar”; co.Open().

2. Methods for manipulating data stored in the database

They provide a set of methods that allow basic operations to manipulate data in the database. Data manipulation language provides a collection of operators that can be applied to validate instances of the data types in the schema and create, modify and extract such instances.

These operators can do the following:
1. Retrieving data from the database (main operation).
2. Inserting new data into the database.
3. Changing existing data.
4. Deleting data from the database.

There are two types of data manipulation languages:
- Procedural data manipulation languages (specific models and hierarchical network) system that allow the user to communicate what data are needed and how they can be exactly retrieved. These applications process information record by record.
- Non-procedural data manipulation language (relational model specific) system that allows the user to communicate what data are needed without specifying how to retrieve data.

These applications process the information sets of records and have the following characteristics:
- provide greater independence of data;
- increase the speed of information processing;
- represent a fourth generation language (4GL - Fourth Generation Language).

Examples of languages that belong to the fourth generation are:
- SQL;
- QBE language;
- generating forms;
- generating reports;
- generating graphs;
- generating applications.

REFERENCES

3. LearnVisualStudio.NET http://www.learnvisualstudio.net resurse educationale gratuite sub forma de filme