Database Systems – Present and Future

Ion LUNGU, Manole VELICANU, Iuliana BOTHA
Economic Informatics Department,
Academy of Economic Studies, Bucharest, Romania
{ion.lungu|manole.velicanu|iuliana.botha}@ie.ase.ro

The database systems have nowadays an increasingly important role in the knowledge-based society, in which computers have penetrated all fields of activity and the Internet tends to develop worldwide. In the current informatics context, the development of the applications with databases is the work of the specialists. Using databases, reach a database from various applications, and also some of related concepts, have become accessible to all categories of IT users. This paper aims to summarize the curricular area regarding the fundamental database systems issues, which are necessary in order to train specialists in economic informatics higher education. The database systems integrate and interfere with several informatics technologies and therefore are more difficult to understand and use. Thus, students should know already a set of minimum, mandatory concepts and their practical implementation: computer systems, programming techniques, programming languages, data structures. The article also presents the actual trends in the evolution of the database systems, in the context of economic informatics.
Keywords: *database systems - DBS, database management systems – DBMS, database – DB, programming languages, data models, database design, relational database, object-oriented systems, distributed systems, advanced database systems.*

1 Introduction

The notion of database system is used in the context of the development of the informatics application with databases. An *informatics application* requires a set of interrelated elements for the collection, transmission, storage, and processing of data with computer. *A Database System – DBS* is a set of interrelated elements, which allows the development and the deployment of a database application. These elements refer to data, software and others resources necessary in the development of a database application. The data are structured and stored on the computer, in the external memory (database) with specific software products (Database Management System – DBMS and application programs) and in a certain work context (legislative framework, organizational framework, equipment, human resources etc.). Thus, DBS involves a great complexity, a lot of components, and a large volume of data. All these aspects result from the structure of a DBS, namely from its architecture.

DBS architecture is a graphical and suggestive representation of the system elements and of the links between them. In the specialty literature are presented various DBS architecture types [12], [8]. We propose a simplified architecture, as well as suggestive and comprehensive, easy to understand and use. Our experience theoretical and practical on DBS, and the research undertaken in this area, helped us to build a components architecture for such a system (figure 1). The main advantage is that any type of DBS architecture can be adapted to the components architecture. Also, trends of development for DBS will generate new components that could fit into the architecture that we have proposed it. Therefore, it results that our architecture has a large portability, flexibility and simplicity. The components architecture gives an idea of the constituent elements of a DBS and of the interdependence between them.

From the proposed architecture of DBS results three components:

1. The data are organized in a database – DB, which include: data collections; the data dictionary containing the data structure, constraints etc.; annex files that refer to the parameters files, index files and so on.

2. *Software* is the component addressed to develop and to explore the database and contains: database management system – DBMS and application programs.
3. *The auxiliary elements* are complementary components of the two above, which contribute to the completion and working of the entire DBS: set of automated procedures (routines) and manuals; legal, administrative and organizational regulations; hardware used; categories of users involved.

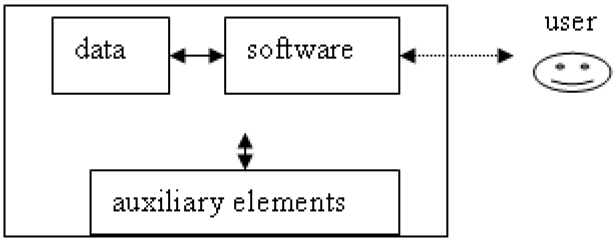

Fig. 1. Components architecture of a DBS

DBS types can be grouped into different categories. These categories are obtained taking into account certain criteria for classification, of which the most important criteria refers to the logical data model implemented. Thus, are obtained two *categories of DBS*, each with subcategories, as we present below.
- DBS that implement the classical logical data models (as shown in the 1st paragraph), in historical order, with the following subcategories: hierarchical, network, relational, object oriented. In such a DBS, will work properly the corresponding DBMS;
- DBS that implement advanced logical data models – is based on classical logical data models which are adapted and developed with new informatics technologies, in order to reach the following subcategories: distributed, spatial, parallel, multimedia, XML etc. (as shown in the 3rd paragraph). This is actually the current trend of development for DBS [2].

2. Database – DB

Organizing data is an important activity in the process of developing informatics systems. The performances of the informatics system depend greatly on how data are organized. In the process of organizing data are followed mainly two major objectives concerning optimizing computing resources: fast access to data and small memory space occupied. Sometimes these two objectives are contradictory. If data access is faster, it is necessary memory space in addition to provide additional information to ensure the access. These additional data can be: links, files with addresses, pointers, index files etc.

The data file contains a collection of homogenous data, which are organized by a given organization technique on an electronic support that can be processed with the computer. There are two levels for the description of data files: logical and physical. The main ways of organizing data files are: indexed, direct, relative, by partitions, multi-indexed, reverse, with links etc. A number of techniques for organizing data files are also used for organizing data in databases. File management is performed by a component of the operating system named File Management System.

The term *database* has appeared first in the conference Development and Management of a Computer-Center Data Base, organized by System Development Corporation in Santa Monica in 1964. However, the definition was made in the technical report of the conference CODASYL, in 1969, which was dedicated to the data management languages. Unlike data files, the novelty are given by the existence of a file that globally (conceptual) describes the data and the links between them. In this way, it is ensured logical and physical independence of data towards applications.

The databases are characterized by the fact that they contain data collections homogeneous and interrelated [19] and that they ensure the logical and physical data independence

towards applications via a third level of data description - the global logical level called also the database schema.
Database management is performed by software called *Database Management System – DBMS*, which is portable on certain operating systems and certain computer systems.
In the databases case, the data access is done quickly by several users at the same time, in various forms and criteria; it is increased the level of protection of the data; it is kept a minimum and controlled redundancy; data are structured according to a data model etc.
The databases and their facilities have evolved from the first type of database to the newest appeared.
Taking into account the development so far and the current trend of databases, we propose a complete definition for the concept of database. *Database - DB* is [13] a whole collection of data, stored in the external memory, with the following characteristics:
- organized on three levels (conceptual, logical, physical - see the three-tier architecture of DBS);
- structured according to a logical data model for DB (hierarchical, network, relational, object oriented);
- consistent, ensuring integrity constraints and data protection;
- with a minimum redundancy, controlled by implementing a data model and by applying a design technique (the technique of normalization for relational DB);
- accessible to more users in a timely manner, so that multiple users can use information from DB whenever they need.

The first databases that appeared were *the hierarchical databases* and *the network databases*. They are based respectively on the hierarchical model and the network model. They are characterized by the three levels of description: logical, physical and conceptual, a minimum redundancy, and the fact that the links between data are done through physical address or through pointers. These databases allow easy access by multiple keys, but also present a disadvantage because of slow updates caused by the physical data links. This is one of the main reasons for which were searched new solutions for organizing data.
The premises of the *relational model* can be found in the concept of the sets/ensembles data model, defined by D. F. Childs in 1968. He indicated that any data structure could be represented through data tables with relationships between them.
The *relational databases,* as a new form of organizing data, are based on the relational model proposed by E. F. Codd in two papers published in the years 1969 and 1970. He defined the relational model [12] through the relational data structure, the operations of relational algebra and relational calculus, and integrity rules or constraints (restrictions) required for maintaining accurate and consistent data.
The relational data model was mathematical fundament and has been a basis for building relational languages and relational database management systems. This model is associated to the normalization theory that optimizes database structure, by eliminating possible update anomalies.
The relational structure has, as basic element, the relation, which is part of a Cartesian product of several domains of data, contains tuples with significance, and has a name. All tuples of the relation should be unique. Representing relations in a two-dimensional table (data table) is easy to understand and use. The relationships between data tables can be logically created through connection codes (primary keys, foreign keys). The relations in a field of activity, mandatory normalized according to defined rules, and the relationships between them, form a relational data structure for that area. This structure is materialized in the database schema, which contains the names and attributes of each table, and the relationships that can be logically established between them. On this relational structure are acting operators of the relational calculus and of the relational algebra.
The Relational Calculus (RC) was proposed by E. F. Codd and it is based on the first-order predicate calculus, which is a field of mathematical logic.
The basic construction in the relational calculus is an expression of the tuple relational

calculus or of the domain relational calculus (depending on the type of variable used). Relational expression calculation is made of: operation performed, variables (tuples or fields), conditions (for comparison, of existence), well-defined formulas (constant, variable, function, predicate), operators etc. The operators used in the relational calculus are: the universal quantifier (∀), the existential quantifier (∃), the conjunctive connector, the disjunctive connector and the negation connector.

The relational algebra (RA) is a collection of formal operations applied to the relations, and it was mathematical fundament also by E. F. Codd. The operations are implemented in relational algebraic expressions, which are the database queries. They are composed of operands and relational operators. The operands are always data tables (one or more), and the result of the relational expression evaluation is only one data table.

Codd's standard relational algebra consists of six primitive operators (union, difference, Cartesian product, selection, projection, join) and two derived operators (intersection and division). Later, were introduced other derived operators (specials) or extensions of the standard RA, such as: the transitive closure, the relation splitting, the relation complement etc. These operators can be grouped into set operators and special operators.

The relational algebra and the relational calculus are logically equivalent: for any algebraic expression, there is an equivalent expression in the relational calculus, and vice versa. The RA is by definition non-procedural (descriptive) while the RC allows procedural and non-procedural searches.

The integrity constraints defined in the relational model represent the main method of integration of the data semantics in the relational databases. The advantage of the introduction of the data semantics into the database, through mechanisms of defining and verifying these restrictions, consists in the easiest way of maintain applications and implement effective physical mechanisms. The two types of restrictions placed in the relational model, the structural and the behavior restrictions, have been studied in terms of possibilities for the verification of compliance and for their power to modeling, so as to be consistent and accurate data into the database.

The relational databases are based on the relational model. They can be consider to be formed from a set of relations (data tables) that can have logical relationships between them, and the data dictionary, in which are described data, relationships, constraints, views etc.

Relational databases present precise advantages in front of the hierarchical or the network databases. Thus, they eliminate the physical links between data (references, pointers etc.) and contain data structures easy to manipulate them, assure an increase degree of logical and physical independence of data towards applications. Relational databases offer new control possibilities for data coherence and correctness, multiples facilities for defining and manipulate data, and allow an increase integrity and security of data and also fast access to the data.

Though, the relational databases present some limits. They offer less support for: multimedia applications, GIS (geographic information system), knowledge-based systems, computer aided design, informatics areas where is working with complex objects. One modality of storing such objects is represented by unconventional data types, such as BLOB (Binary Large Object), which are allowed in all the relational databases. In the relational databases, these objects are considered entities with no internal structure, therefore isn't any possibility of finding or accessing their elements. These deficiencies led to the introduction of object-oriented technology concepts in the area of databases, leading to the object-oriented model, and respectively to the object-oriented databases.

At the base of the *object-oriented data structure* are the following concepts: object, object class, hierarchies of object classes, inheritance, encapsulation, persistence, polymorphism etc.

The objects are basic structures that include data structures and methods, and are grouped in classes or types of objects. The object classes are related by inheritance and form class hierarchies. Data structure contains complex objects, made up of simple components, each with its own attributes and behavior. The operations of the object-oriented data model can be grouped in: getting and sending messages, selecting the appropriate methods, updating methods, updating classes etc. In the object-oriented data model, the integrity constrains are deducted from the definition of structure and operations, and they are: the encapsulation constraint, the constraint on compliance with the protocol specified by the definition of the class, the unique object identifier constraint etc.

The object-oriented database allows storing and selecting data through object-oriented technology. It contains classes of objects among which there are different hierarchical links or another type of link and which complies with the rules of creation and usage of the objects.

These databases have the advantage of better reflecting the real world that consists in complex objects, of different types, which can be decomposed in other objects and over which can act events to change their status. The access at database objects is much faster because of the addressing mode based on pointers. Also, object-oriented databases allow the definition reusability, which increase efficiency in the creation and use of the database. They are used in the domains where there is no need for complex objects and relations to be broken and then to be reassembled for use.

In the recent years was founded a new data model, namely *the multidimensional model*. This model represents the data as a data cube. The data cube allows modeling and visualization of data in multiple dimensions. A data cube is a set of information, organized and presented in a multidimensional structure with a set of dimensions and measures. The data cube provides a mechanism for querying data with a response time very short. Each data cube has a schema, which contains the facts table that is the source of the cube measures, and the dimension tables that are the sources of the dimensions. The most popular multidimensional models are: star schema, snowflake schema, constellation schema. Multidimensional model is the basis of defining data warehouses as a way of organizing data. In the widest sense, a *Data Warehouse (DW)* is a complex database that is maintained with data from internal and external sources of the organization. Data from source systems are extracted, cleaned, transformed and stored in special data warehouses, in order to support decision-making processes [14]. DW is a collection of subject-oriented data, integrated, historical and non-volatile, which is supporting the process of making decisions [18]. This vision of DW focuses on their role in the decision information management, maintaining in this way a high level of generality.

DW has the following characteristics [16]: allows the access to organizational data, the data are consistent, and can be combined and separated according to each dimension or every aspect of business. DW will have attached a software product that provides a set of tools for data query, analysis and presentation. This is where the data used are published, and the quality of these data contained in DW will be a prerequisite for business reengineering.

IBM Company uses for data warehouses the term: Information Warehouses. Moreover, in the specialty literature are used simultaneously the two terms of data warehouse: Data Warehouse and Information Warehouse.

The purpose of a data warehouse is to develop a data repository that will make available operational data in a form acceptable to support decisions and for other applications [14]. In terms of area coverage, there are three models of DW: enterprise data warehouse, data mart, virtual data warehouse.

The Enterprise Warehouse collects all the information about topics related to the whole organization [17]. It provides an extensive amount of data (Terabytes). Usually, contains detailed data, but can include also aggregated data. Enterprise data warehouse can be implemented on traditional mainframes, on super-servers UNIX or on platforms with parallel architectures. This requires large expenses for modeling and years for design and execution.

The Data Mart contains a subset of the data volume of the organization, which is specific to a group of users [18]. The domain is limited at specific subjects. The data contained in the data mart are usually aggregated. Currently, the data marts are implemented on cheaper departmental servers, which are based on UNIX or Windows NT. The cycle of implementing a data mart is rather measured in months. As such, a data mart can be considered a part of a data warehouse, easier to build and maintain and less expensive.

The Virtual Warehouse is a set of views on the operational databases. For the efficiency of query processing, some of the views of aggregation can be materialized. A virtual warehouse is easily to build, but requires additional capacity on the database servers.

3. Database Management Systems – DBMS

At the beginning of this article we presented the part that Database Management Systems – DBMS play in the by component architecture of a Database System. Based on that we devised a series of definitions for the DBMS [9], listed below:

A DBMS is a complex ensemble of programs that provide an interface between a database and its users. A DBMS is the software component of a database system that interacts with every other component, ensuring the connection and independency between the system's elements. Taking into account all these definitions, a software product is a DBMS if all of the following *assertions* are simultaneously true:

- it's a system – an ensemble of interconnected programs that collaborate with each other to attain a shared purpose – creating database applications;
- it manages data organized in the external memory, according to a logical data model for the database;
- it achieves the objectives and functions of a DBMS.

Hence, a DBMS must, at a minimum, satisfy two conditions: to implement a logical data model for the database, to incorporate at least a programming language as well as interfaces/instruments to optimally manage the data

The relational DBMS are the ones that have been mainly used in the last 30 years – they implement the relational data model and at least a relational programming language – usually SQL. This type of DBMS has greatly evolved in the past 10 years, adding new features based on applying new technologies – object oriented, distributed, Business Intelligence etc. This trend was maintained up to present, when DBMS become more than just complex database software, they became an infrastructure for databases.

The DBMS role

Using the definition already given for a DBMS and some others that exist in the specialized literature [12], the intent of such a software system becomes clear. We have delimited the role of a DBMS in a database system context and designed a suggestive diagram (figure 2). So, the *role* of a DBMS is to:

1. *define and describe* the structure of a database through a specific intrinsic language - Data Description Language - DDL, corresponding to a certain logical data model;
2. *load-validate* the data in the database, respecting some integrity constraints enforces by the data model in use;
3. *make access* to the data for different operations (consulting, interrogations, actualization, reports editing) using the data model operators;
4. *database maintenance* using specialized instruments (editors, shells, browsers, translators etc);
5. ensure *database protection* namely the security and integrity of data aspect.

The examination of this last diagram leads to the conclusion that, for a DBMS to work properly, it has to have at its disposition a conceptual schema of a database. This schema is built [8] based on the real world, delimiting the domain of interest. This domain has to be the subject of a study/investigation activity – to identify the activities, resources and transformation-objectives – then the base activity has to be covered – that means a conceptual modeling to emphasize the applications requirements (an entity-relationship diagram will emerge). Based on the obtained results, the design of the database step will be covered, applying specific techniques, such as normalization for a relational DB. The result is a conceptual schema of a database – that contains, in a graphical form, all the entities, their characteristics and connections. Up until this moment the design team was in charge. From now on, the DBMS takes charge – it can work like in the last diagram. To this end, the DBMS consists of a series of software components called by the DB realization team.

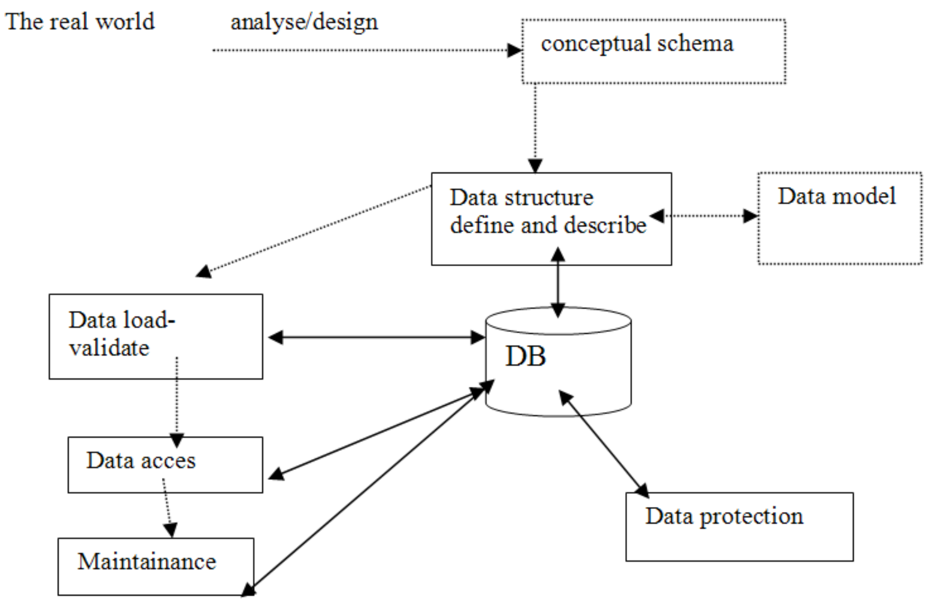

Fig. 2. The Roles of a DBMS

To fulfill its goal, a DBMS has to use all its components on the database. The current trend is that the DBMS's role extends more and more, taking on many more features from one version to another, becoming more of a database infrastructure.

The DBMS functions

The entirely of a DBMS components ensure its role through four obligatory functions. To fulfill the four functionalities, each DBMS supplies a series of activities on the database, using its components. The grouping of these activities on functionalities has a relative nature – taking into account the complexity of DBMS, the functionalities offered, the employed programming languages and the method of implementing the data model. Different DBMS have distinctive features, based on the implemented data model, identified through specific operations and activities. Despite these characteristics, there are some functions available for all the DBMS categories (as shown in figure 3). These are some basic functions, and if a software system does not provide them, it cannot be considered a DBMS. These functions are: the data description, the data manipulation, the data use, the data management. All of them are carried out through operations on data organized in a database. There is a tendency to keep the functions of a DBMS the same, even if the systems keep expanding and including new features.

A DBMS, through its functions, allows the authorized users access to the database, as specified in our definition of a DBMS. A short presentation of each of the four basic functions of a DBMS follows.

1. *The data description*

A DBMS allows the definition of the database structure, using the Data Description Language – DDL. Defining the data can be realized at a conceptual, logical and physical level. The following items need to be described: attributes (fields) in the database structure, connections between database entities, criteria to validate the data, methods to access the data, aspects pertaining to providing referential integrity. The tangible objective of this function is the database schema, memorized (committed) in the database dictionary. This function was greatly automatized, and now a DDL has a small number of commands. The DDL is specific to each DBMS, but it always produces the description of data according to the elements of the data model that specific DBMS uses. At the end of this function, the database entities exist as files in the DBMS, but they do not contain data, only the database structure (database schema).

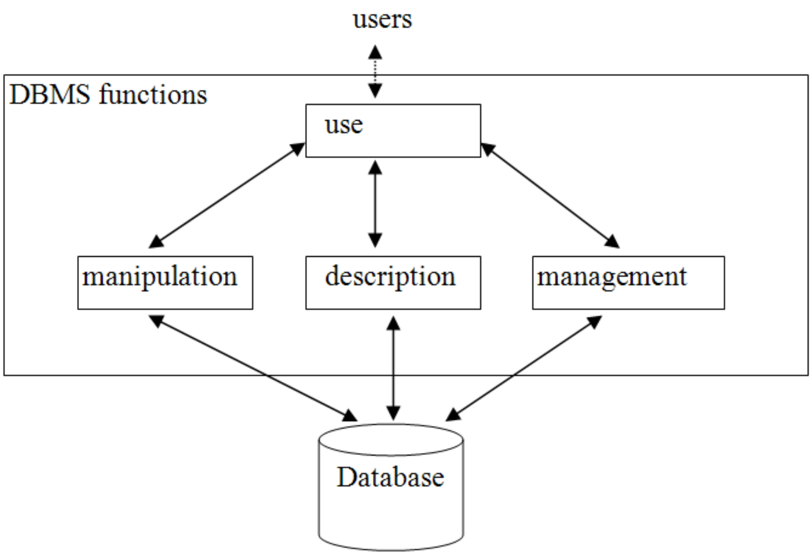

Fig. 3. DBMS functions

2. *The data manipulation*

The manipulation function is the most complex one; it delivers update and query of data from the database, using the Data Manipulation Language – DML. The following procedures can be performed on the data: load, update, processing, and query.

The data load into the database is executed through automated or programmed operations that ensure the necessary validation criteria.

The update of a database consists in operations of adding, modifying and deleting the records. The same validation criteria used for loading the data have to be used during the adding and modifying of the records. Updating is possible only after authorization, by ensuring data protection to conserve the database coherence.

The data processing is carried out through selecting, ordering, grouping, un-grouping of the database entities. These are, usually, computations made prior to querying the database. Many of these data processing operations are accomplished with the help of some of the operators of the data model implemented by the DBMS.

The data query / interrogation consists of operations like displaying (on screen or on paper), database browsing, output editing. The outputs can be final or intermediary, and can be obtained on several information supports: on screen, on paper, on a magnetic medium, on an optical medium. They can be presented in many ways: bulleted lists, reports, graphs, images, sounds, video and can be obtained using different search criteria.

A DML can use a host language or its self language. DML with host language are developed through adapting some universal programming languages (like COBOL, Pascal, C, Java, etc) to the DBMS requests. This way, the power of an universal programming language is combined with the data query requirements (e.g. Oracle). The ones with self language are developed in a specific language, capable of reuniting the power of procedures with querying of a particular database type (e.g. Visual FoxPro). For the query activity there are specialized query languages that can be included in the DML or that can exist as-is.

3. *The data use*

This function provides an assortment of interfaces needed to ensure the communication of all the users with the database. To implement this function, the DBMS has to provide facilities for several user categories: end-users, expert-user, managers.

The end users are the major category of users – recipients of the information stored in the database. The DBMS allows them to use nonprocedural languages and other database querying facilities (generators, utilities) in a simple and interactive manner. These users do not have to be familiar with the database structure and / or programming languages, the DBMS helps them to interactively use the database through – menus with suggestive options, windows, templates, wizards, comprehensive help (tutorials).

The expert users in informatics create database structures and complex procedures to explore the database. DBMS provides these users with the DDL, the DML and interfaces with universal programming languages – that vary in complexity and capacity from one DBMS to another – presenting nonprocedural and procedural items to the expert user. Using them, the expert user describes the database schema and complex ways to manipulate data. To create a database, the DBMS will provide to the user CASE (Computer Aided Software Engineering) elements that help in different design steps.

The DB manager users have an important role in the optimal operation of the system. Due to the importance of this category, the DBMS has a distinct function to serve them.

4. *The data management*

The administration function is complex and can be performed only by a database manager. Such a user, which has a rich background in analyzing, design and programming, organizes and administer a database in all of its design stages. He sets-up the database using a particular methodology, creates the conceptual database schema, and coordinates the database design. To achieve all these tasks, the DBMS provides a series of CASE elements and specialized utilities.

In the operational stage of the database, the administrator has to authorize data access (set-up accounts, passwords etc), to rebuild the database in case of accidents (through journalization or copies), to efficiently use the storage space in the internal and external memory (through organizing, optimization routines), to provide a series of statistical analyses for the database (number and type of users, number of logins, number of updates etc). For each and every of these activities, the DBMS provides a mechanism or a working technique.

In the case of a network setup, using distributed databases, the DBMS has many components dedicated to the database administration, because the database is complex and the data are distributed on all of the computers in the network and there are many users of different types.

Mainly to serve the administrative function, but also helping the other three functions, a DBMS provides protection of the database, under both aspects: security and integrity.

The DBMS architectures

From their appearance up to present, DBMS have known a great variety, and therefore it is difficult to give a unique architecture, valid for all their types, because are frequently appearing features from one system to another. There are concerns about the standardization of DBMS architecture, which seeks to define a general framework. Among them, two reference architectures of DBMS are proposed by the researchers group of CODASYL and ANSI / SPARC [15]. The trend in recent years is that the DBMS architecture has evolved to a configuration with three components (as shown in figure 4) – kernel, interfaces, and tools. This situation is encountered at the latest versions of commercial systems. Another trend is represented by distributed and object-oriented architectures of DBMS [13], which are more commonly used in network computers and in new types of applications. Since components / levels architecture reflects the current trend of DBMS structure, we further present it.

Starting from the specialty literature and studying the current trends, we have identified a modality of grouping the DBMS components and their graphic representation on three levels, and therefore we have designed the components/levels architecture of DBMS. The main advantages of our architecture are the following: it is very simple, easily understood, easily implemented, and it is portable - any kind of DBMS architecture can be adapted to the three levels architecture.

The components/levels architecture of a DBMS. In defining a DBMS we have shown that not every software product that manages data in the external memory is a DBMS, but only one that meets certain conditions. Result that DBMS contains a number of components, which are interfaces and software tools that are designed to meet specific system functions.

The various components in different types of DBMS - each came with one or more proposals for architecture - can be placed, sometimes questionable, in one of these three levels. The architecture levels of the above can contain the following components of a DBMS:

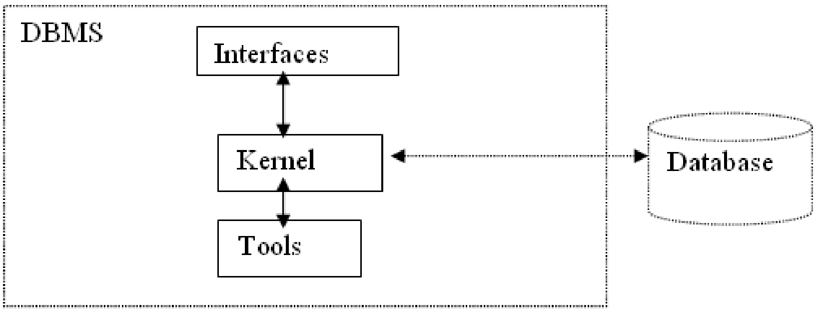

Fig. 4. The components/levels architecture of a DBMS

- *the kernel* contains DDL, DML, and the mandatory *components* in the minimal kit of DBMS. This component is designated to analysts, programmers and DB administrators;
- *the interfaces* are composed of: generators of various types - menus, forms, reports etc.; CASE elements - Computer Aided Software Engineering; interfaces with the universal programming languages; interfaces with other systems etc. The component is intended for all categories of users: end-users, experts;
- *the tools* consist of: editors, browsers, shells of various types. The component is designed, essentially, for the database administrator, but also for other categories of users.

Conclusion. We believe that the components/levels architecture is simple, but complete. Therefore:

- standardized architectures - CODASYL and ANSI can be made on three levels;
- new database technologies (e.g. object-oriented technology) have led to the emergence of new types of DBMS. The architectures proposed for them can be adjusted on the three levels;
- new informatics technologies (such as multimedia, Internet etc.) have interfered with the database technology, leading to appropriates evolved DBMS. For them were proposed architectures which can be adapted to the structure on three levels.

4. The advanced DBS

As we have shown earlier in the text, at *present*, the most widely used database systems are the *relational* ones. The relational DBS have permanently developed new facilities, a series of remarkable optimizations and have continually adapted to the IT context, which, as we all know, has an extraordinary dynamics. The databases technology has interfered with other information technologies, resulting in *new hybrid* types of database systems. These are based on one of the fundamental data models for databases (hierarchical, network, relational, object oriented) and are extended with new facilities [2]. Thus, the so-called *advanced DBS* were created, among which we shortly present a few of them.

The parallel DBS are the result of the integration of the database technology with the technology of parallel processing on calculating systems and in computer networks. Concerning such a system, we are *interested* in:
- the necessary operations of parallel processing – on data, query requests, transactions, concurrent access;
- the available calculation resources, which means the partitioning of internal memory, of external memory, of all the available calculation resources.

The mobile DBS refers to database applications destined to mobile equipments – mobile phones, PDA, POS etc. – connected to portable microcomputers – Laptop, Notebook, Palmtop etc. - and to a communication network. *The main features* of the mobile DBS are the following: the users are connected via network and obtain a short response time; the communication cost increases function of the in/out operations and depending on the internal memory processing operations; the user localizations changes permanently; the use time is limited (batteries etc.); one cannot work with centralized transactions, only with distributed ones.

The spatial DBS - using the database technology - resulted by integrating geographical systems (with a memorizing map and associated information) with computer-aided design systems (with the information stored for assisting a design process). The *main features* of the spatial DBS are:
- the spatial data are a collection of multidimensional data, lines, polygons, cubes and other geometrical objects. They can be *point data*, when a point is completely characterized by its location in a multidimensional space, *region data*, when the data are characterized by localization and destination;
- the spatial queries are queries made on spatial data. Here are a few types of spatial data: *of a certain rank* (all the villages in Romania), *of proximity* (three neighboring towns in Moldova), *of junction* (pairs of towns in Romania, situated at a distance of 50 km from each other);
- the spatial index is the index ordering applied on spatial data and it can be done by several techniques: spatial curves, distribution files, R trees etc.;
- the applications types of spatial DBS: GIS – Geographical Information Systems, CAD – Computer Assisted Design, GPS browsers etc.

The multimedia DBS are stores and processes, within databases, classical data (texts, graphics), as well as multimedia data (image, audio, video). *The main features* of the multimedia DBS are: it stores large dimension data (Gigabytes); accepts similar query for images, audio, handwriting; accepts continuous media data, such as sound and video; accepts different data formats (JPEG - Join Picture Experts Group, MPEG - Motion Picture Experts Group etc.); the types of DBS multimedia applications are those which presuppose queries based on content, the use of individual large objects, a great necessity for video data.

The advanced Decision Support Systems - DSS are complex software products which use online data for justifying decisions and for assisting the decision-making process.

The main features of a DSS are: complex queries; classical statistical analyses, by using specific models; Data Mining for the automatic discovery of certain rules and patters (information) using the available data; the processing of large data volumes (Data Warehouse, Data Mart), by using special technologies; they can be used for all types of decision-making problems: structured, semi-structured, unstructured; it has to be included in the integrated information system of the organization; it includes data, as well as models, organized in databases; it offers support for the decision-making activity, but it does not replace it.

The distributed DBS is included in the larger concept of distributed systems. They are used in several IT domains: database systems,

computer networks, operating systems etc. Nevertheless, all the distributed systems, irrespective of their type, have a few common features and objectives: support for resource partitioning, openness, competition and parallelism, scalability, tolerance to accidents, transparency.
The distributed DBMS is the result of the integration of the DB technology with the computer network technology by extending a DBMS with data communication and management facilities in the network. DDBMS manages several local DBs, integrated by a communication network; thus, the user, no matter his location within the network, perceives a single DB. The researcher C.J.Date established 12 rules, according to which it can be established if a DBMS is distributed and to what extent. The main idea resulting from the rules is that data distribution must not affect users in any way; in other words, DBMS must ensure a total transparency of data distribution.
Data distribution is ensured by a DDBMS, by implementing some specific techniques: fragmentation, replication, mixed, loading.

Other information technologies implemented in advanced DBS

Business Intelligence is an information technology which deals with organizing and running an enterprise, as well as its management, based on the solutions of the advanced IT.
The BI domain is included in the tendency of transition from the industrial society to the information and knowledge society [3].
Business Intelligence presupposes the use of all the data available to a firm, by means of computers, with the purpose of improving the decision-making process. This objective requires access to the data, their analysis and finding new possibilities for using them, that is a set of information technologies, used in the business making process.
Business Intelligence refers to the capacity of transforming existent data into useful information, which can provide a wide range of perspectives, especially new perspectives, on the business world at present and which can offer an idea on its future development.
BI requires the capacity of processing a great number of entries, of performing complex calculations and of aggregating the data into significant summaries. The database designed for the BI must be optimized for reports. Such a database will often store very large amounts of historical data and it can be much larger than a transactional system. The DB requirements for the transaction processing are presented comparatively to the BI requirements (Table 1) in the following table:

Table 1. Transaction processing vs. BI

	Transaction processing	**BI**
1. Purpose	Automatization of a repetitive process	Reporting, data analysis and discovery
2. Designing	Minimal and controlled redundancy, dynamic calculations	Introduced redundancy, flat data structure, complex calculation
3. Data storage	Discreet transactions, current data, an application	Transaction summary, historical data, multiple integrated applications
4. Access to data	Updating, fast queries	Only queries and average response times

The technologies employed for BI systems are described next. Today, only an estimate of about 12% of information technology solutions were not designed based on the database systems and the forecasts are underlying a further drop of this percentage in the near future.
Data Warehouses and ***Data Marts*** solve the issues concerning scattered data sources and incompatible purposes between transactions processing and BI applications. A data warehouse's purpose is to supply central data storage for one or numerous transactional systems, thus achieving a single, integrated and consistent data source. The *data warehouse* is designed to optimize the process of reporting for a large number of database records [7]. It involves numerous data retrievals and

very little, if any updates. The development of enterprise data warehouses was long and complex and its practitioners had to employ a new approach: the development of smaller, consolidated warehouses, known as *data marts*. Thus, storing data in smaller quantities provides the opportunity of high accuracy and fast reporting, attainable in a shorter development cycle [10]. These facilities, once integrated into DBMS instruments, meet all the requirements of a business, since they rely on such a large and diverse array of data, with outstanding retrieval capabilities.

Extraction-Transformation-Load – ETL takes into account the creation of a data warehouse based on various data sources. Developing a single, consistent data storage gathered from multiple systems requires *data cleaning*. Also, the numerous data sources may require data transformation to a common, unique format, prior to feeding the data warehouse. An ELT instrument give the possibly to define business rules by using: a graphic interface, standard communication interfaces (e.g.: ODBC, JDBC etc).

OLAP – ***OnLine Analytical Processing*** employs multidimensional analysis in order to achieve flexibility, yet maintains a steady performance level. Through this approach, data is perceived as a cube concept. This cube consists of quantitative values (known as *measures*) and descriptive categories (known as *dimensions*). *Utility* for companies [4] will benefit from:

- top management's ability to provide a better analysis of own data, in order to take the best decisions;
- on data analysis, searched factor is always known;
- possibility of multidimensional cloud grouping data search;
- capitalizes on the classic statistical analysis experience, developing new techniques and superior methods.

Data Mining – DM takes BI one step further than OLAP, and it can be stated that it is complementary with OLAP for a number of *reasons:*

- with OLAP technology, the user is actively engaged in data exploring, while in DM, information defines itself without being addressed;
- with OLAP searched items are known, in DM they are not – they are discovered;
- OLAP capitalizes on the classic statistical analysis developing new techniques and methods, while DM capitalizes on artificial intelligence which is enhancing new discovery methods;
- both technologies are searching cloud grouped values into a multidimensional environment, but with different approaches.

A few *characteristics* of DM are:

- it builds upon computation sheets (table computation) based software experience, developing the concept furthermore;
- data analysis and the learning process are achieved through numerous information technologies: artificial intelligence, statistics, mathematics etc;
- it handles the exceptions from the rules;
- employs complex searching methods to identify data patterns and groups;
- it extrapolates and builds upon known or somewhat known cases;
- it always learns and presents a solution with a certain degree of guarantee;
- it's using a vast array of search and extraction algorithms: different tree types, neuronal networks, random search, probabilities, forecasts etc;
- it can identify unforeseen tendencies in consumer behavior, thus enhancing future behavior patterns;
- main target area: marketing and publicity: marketing campaigns to promote goods and services, strategies for the mid and long-term developing of the company;
- establishing basket of goods; data extraction improves with the growth of data amount and requires high stocking quality for useful results.

Example of DM based software products included in database platforms: Oracle 10g Miner, DB2 Data Mining, SQL Server 2008 Data Mining.

The Java platform, powered and promoted by Sun Microsystems USA, derived from the developing of the Java programming language in 1995 as a solution to adapt universal

programming languages to the new, Internet based environment.
Later on, new Java based products were launched, thus creating what we know today as the Java platform: the Java programming language, JavaScript, Servlets, Java Server Pages – JSP, Java Database Connectivity – JDBC, Java Beans, Enterprise Java Beans – EJB, Business Components for Java - BC4J, Java to Enterprise Edition - J2EE, SQL Java – SQLJ etc. Latest types of DBMS have already fully/in part implemented this technology. Tendency [5] of the market is that more DBMS will implement this Java technology, in a certain measure.
Grid computing represents a new technology which grew from the idea that computing technology around the world isn't employed to its fullest [6]. Most of the times, computing systems are difficult to change, costly to operate and develop. Changes within the organizations emerge all the time, the need for information is higher and higher, therefore adaptation must be accomplished quickly and effortless in order to stay competitive.
The demand for performance is continuously rising, while budgets may remain unmodified and therefore organizations develop their own servers or purchase more powerful systems. An actual solution to these issues is a new kind of approach, the Grid Computing – GC. For application users, through GC technology *it doesn't matter anymore* were data are stored, were applications are stored, which computers are processing the search queries, or what resources are used over the network. GC represents coordinated use of multiple smaller servers which act as one very powerful system.
For the first time, GC technology is adapted to database by the Oracle Company, through Oracle 10g. We think that the main information technologies that through integration made it possible for the latest Oracle product version to be released were: GC, Intranet, Internet, multiple servers (Mail, network applications, data, DBMS etc.), NC architecture, Business Intelligence, DBS.
Technology break through took place because of these information technologies, because of the creation of more and more powerful and cheaper components and because of integration of technologies (hardware, software, data, communications). Most important features of GC are: virtualization, dynamic resource supplying, automatic system adaptation, unified and efficient management.
Through GC technology, taking into account its applicability, the Enterprise Grid Computing – EGC concept was conceived. This concept assumes that several computers within an organization should run and *work together* as one integrated system. EGC needs a software program that smoothly enhances the efficient accessing of multiple servers, permits modular data storage and allows on request storage capacity increase.
Web technology is extremely business oriented due to the rapid World Wide Web development and constitutes an almost limitless market, offering a huge source of information. The components of the web technology, for dynamic pages work environment, can be structured into two *categories:*
- description of the application – application's interface, which means that the user is accessing, from a certain point of the network, the web browser and sends queries. Thus it is possible to access the services from a Web Server.
- processing part - logic of application, that means that services are offered through Web Server resident program modules and realized through different technologies (CGI modules, Java platform products etc.). The resident program modules are employed for accessing and processing the databases to obtain results.
In modern DB applications there is the tendency of employing a web technology based user interface and therefore DBMS posses one or more such components.

Conclusions

If we should summarize the current development trend of databases, only in some words, they should be *integration* and *optimization*.
The integration is carried out on different le-

vels, but what we outlined in particular was the integration of databases technology with other informatics technologies. The DBS tend to be differentiated by the new informatics technologies integrated, and the transformations that occurred in this process are considered for optimization [11].

The optimization aims at efficient use of computing resources: time and space. The facilities offered by DBS aim at the increasing optimization, for all the aspects and this trend are maintained for the future. Thus, for example, an older approach – the database machine - is rediscovered and promoted [1] as a future solution for special performances of the databases: speed may increase 10 times!

References

[1] D. Baum, "Launching performance," *Oracle Magazine*, vol.XXIII, No.1/2009.

[2] A. A. Simanovsky, "Data Schema Evolution Support in XML-Relational Database Systems," *Programming and Computer Software*, vol. 34, no.1/2008, pp. 16–26

[3] M. Velicanu and G. Matei "A few implementation solutions for Business Intelligence," *Informatica Economica* no. 3/2008, pp. 138-146.

[4] I. Lungu et al. *Soluții informatice de asistare a procesului decizional si pentru dezvoltarea managementului bazat pe cunostinte in institutiile publice,* Grant PNII (Idei) CNCSIS, 2007, 2008.

[5] M. Velicanu and M. Anghel, "EJB components from Oracle platform in web applications," *Economic Computation and Economic Cybernetics Studies and Research*, nr.1-2 /2008, pp. 185-192.

[6] M. Velicanu and S. Ionescu "Monitorizarea sistemelor informatice integrate folosind Grid Computing," *Revista Studii şi Cercetări de Calcul Economic şi Cibernetică Economică*, no. 4/2008, pp. 81-94.

[7] M. Velicanu and G. Matei *Database versus Data Warehouse,* Baza de date internațională SSRN eLibrary Stats, iunie 2007, www.ssrn.com.

[8] M. Velicanu. *Baze de date prin exemple*, Ed. ASE, Bucharest, 2007.

[9] M. Velicanu. *Dicționar explicativ al sistemelor de baze de date*, Ed. Economică, Bucharest, 2005.

[10] M. D. Solomon, "Ensuring a Successful Data Warehouse Initiative ," *Information Systems Management*; 2005, pp. 26,.

[11] J. M. Hickand and J. L. Hainaut *Database application evolution: A transformational approach,* 2005

[12] J.Date. *An introduction to Database Systems*, Addison Wesley, 2004.

[13] M. Velicanu, I. Lungu, M. Muntean, and S.Ionescu. *Sisteme de baze de date – teorie şi practică*, Ed. Petrion, Bucharest, 2003.

[14] E. Turban and J. Aronson *Decision Support Systems and Intelligent Systems,* Prentice Hall International, Upper Saddle River, New Jersey, 2001.

[15] M. Velicanu, I. Lungu, C. Bodea, C. Ionita and G. Badescu. *Sisteme de gestiune a bazelor de date*, Ed. Petrion, Bucharest, 2000.

[16] R. Kimball, L. Reeves, M. Ross, W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit,* John Wiley & Sons, New York, 1998.

[17] J. Ryan. *Building and deploying an enterprise data warehouse,* White Paper, 1999.

[18] W. H. Inmon. *Building the Data Warehouse,* New York, John Wiley & Sons, 1996.

[19] I. Lungu et al. *Baze de date. Organizarea, proiectare, implementare,* Ed. ALL, Bucharest, 1995

Ion LUNGU is a Professor at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. He has graduated the Faculty of Economic Cybernetics in 1974, holds a PhD diploma in Economics from 1983 and, starting with 1999 is a PhD coordinator in the field of Economic

Informatics. He is the author of 22 books in the domain of economic informatics, 57 published articles (among which 2 articles ISI indexed) and 39 scientific papers published in conferences proceedings (among which 5 papers ISI indexed and 15 included in international databases). He participated (as director or as team member) in more than 20 research projects that have been financed from national research programs. He is a CNCSIS expert evaluator and member of the scientific board for the ISI indexed journal Economic Computation and Economic Cybernetics Studies and Research. He is also a member of INFOREC professional association and honorific member of Economic Independence academic association. In 2005 he founded the master program Databases for Business Support (classic and online), who's manager he is. His fields of interest include: Databases, Design of Economic Information Systems, Database Management Systems, Decision Support Systems, Executive Information Systems.

Manole VELICANU is a Professor at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. He has graduated the Faculty of Economic Cybernetics in 1976, holds a PhD diploma in Economics from 1994 and starting with 2002 he is a PhD coordinator in the field of Economic Informatics. He is the author of 18 books in the domain of economic informatics, 64 published articles (among which 2 articles ISI indexed), 55 scientific papers published in conferences proceedings (among which 5 papers ISI indexed and 7 included in international databases) and 36 scientific papers presented at conferences, but unpublished. He participated (as director or as team member) in more than 40 research projects that have been financed from national research programs. He is a member of INFOREC professional association, a CNCSIS expert evaluator and a MCT expert evaluator for the program *Cercetare de Excelenta - CEEX* (from 2006). From 2005 he is co-manager of the master program *Databases for Business Support*. His fields of interest include: Databases, Design of Economic Information Systems, Database Management Systems, Artificial Intelligence, Programming languages.

Iuliana BOTHA is a Pre-assistant Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2006 and the Databases for Business Support master program organized by the Academy of Economic Studies of Bucharest in 2008. Currently, she is a PhD student in the field of Economic Informatics at the Academy of Economic Studies. She is co-author of one book, 2 published articles (one article ISI indexed and another included in international databases), 5 scientific papers published in conferences proceedings (among which 1 paper ISI indexed). She participated as team member in 3 research projects that have been financed from national research programs. From 2007 she is the scientific secretary of the master program *Databases for Business Support* and she is also a member of INFOREC professional association. Her scientific fields of interest include: Databases, Database Management Systems, Design of Economic Information Systems, Grid Computing, e-Learning Technologies.