International Journal of Research in Engineering, Science and Management
Volume 5, Issue 11, November 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

A Case Study of RDBMS and OODBMS: Importance in Business

Nutan Govinda Bhoyar*
Student, Department of MCA, Ajeenkya D. Y. Patil University, Pune, India

***Abstract*: A comparison of various data stored in databases utilizing relational database approach and object-oriented database strategy is shown in this research. The development, deployment, and maintenance of vast and complicated data utilizing RDBMS and OODBMS are described in the study's methodology. The size of the web is growing dramatically right now, and so are mobile applications, technologies, social media, etc. The quick development of these technologies results in a variety of structured and unstructured data types with various security concerns. Take fingerprint databases as an example. These databases are distinct and managed by different technologies, such as MySQL, and different data is produced using different methods. Additionally, this article covers the pros and advantages of the relevant databases and identifies the optimum database system for an organization. Although it is very tough to determine which database is best suited, some are more advantageous and beneficial than others.**

***Keywords*: RDBMS, ODBMS, file-oriented strategy, complexity, inheritance, object identity, object-oriented architecture, evolution of object-oriented database.**

1. Introduction

Data are the foundational resource of the company in computerized information systems. Therefore, for an organization to function effectively, proper data organization and administration are needed. Database management system is nothing but storage of data and the management of that data in computer information system. Any organization needs accurate and trustworthy data for better decision-making, protecting data privacy, and effectively managing data. Examples include making a deposit or withdrawal from a bank, booking a hotel, airline, or train, buying groceries, and accessing databases. Data can be facts or figures which have implicit or explicit meaning. It can be also called as the representation of facts, concepts or instruction which can be easily understandable and can be easily processed. Data can be represented in alphabets (A-Z, a-z), digits (0-9) and using special characters (+,-.#,$, etc) e.g: 25, "ajit" etc.

Information is the processed data that serves as the foundation for decisions and activities. Information can be summed up as data that has been arranged and categorized to produce valuable values.

A. Drawbacks of a File Oriented Strategy

1) Data inconsistency and redundancy

The same data may be written in multiple files. This redundancy results in higher expenses for storage and access. It can lead to inconsistent data that are different copies of the same data can present in multiple places. For instance, a modified client address may be reflected in a single file, but not elsewhere in the system.

2) Data access issues

The conventional file management mechanism does not retrieve data effectively or conveniently.

3) Data isolation

It can be challenging to retrieve the pertinent data because it is dispersed over several files and may be in different formats depending on the new application programmers being used.

4) Integrity problems

The developers impose the validation of the data in the system by adding the appropriate code in the various application programs. However, when new constraints are added, it is difficult to change programs in order to ensure compliance.

5) Atomicity

When a transaction fails due to power outages, network problems, etc., it is challenging to guarantee atomicity in a file processing system. (Atomicity: Whether or not a transaction's operations are appropriately recorded in the database.)

6) Concurrent access

In the file processing system, accessing the same file for a transaction at the same time is not possible.

7) Security issues

The file processing system has no security to guard against unauthorized user access.

A database is an organized collection of associated data from a formatted organization that is shared by several users. The essential characteristics of data in a database are: It must be neatly organized. It must be related. It must be easily available in a logical order, A new strategy was necessary to get around a file system's restriction. A database strategy resulted from this. A database is a long-term collection of information with logical connections. Initial efforts focused on offering a centralized database of data. A database's nature is one of self-description. It includes more than just data sharing and organization-wide integration into a single database. A small

*Corresponding author: nutanbhoyar11122001@gmail.com

database can be managed manually, but a large database with many users is challenging to keep up with. A computerized database is helpful in that situation. The following are some benefits of database systems over conventional, paper-based recordkeeping methods:

The machine can retrieve and edit the data more quickly than a human being, so there is no need for a lot of paper files. Accuracy: Requirement-specific information is acquired with accuracy, eliminating much of the tedious manual file maintenance.

A database management system refers to a group of programmers for specifying, creating, maintaining, and manipulating databases. It consists of a collection of connected data. It must provide the ability to define the database's structure and define the permissions for authorized users. The DBMS must include features like record insertion into databases, data updating, data deletion, and data retrieval. Multiple users must share data items with the DBMS while ensuring data consistency. It needs to prevent unauthorized users from accessing the database. If the system malfunctions for whatever reason, the DBMS must enable data base recovery.

Database Management System (DBMS) is software for storing and retrieving users' data while considering appropriate security measures. It comprises of a number of software applications that operate on the database.

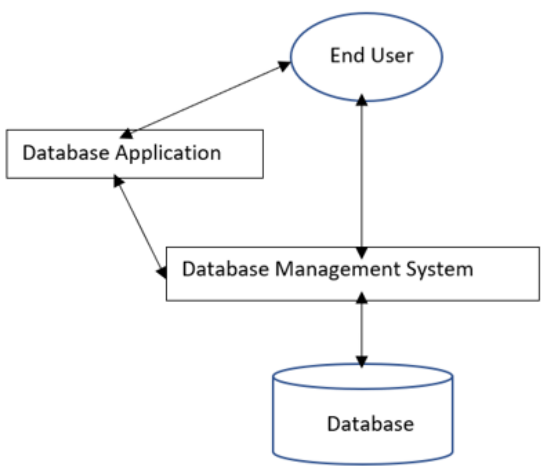

Fig. 1. Components of Database Management System

The DBMS accepts the application's request for data and notifies the operating system to give the requested data. In large systems, a DBMS helps users and other third-party software store and retrieve data. DBMS allows users to create their own databases as per their requirements. The user of the database and other application applications are included in the term "DBMS." It offers a connection point between the data and the software program.

B. Relational Database Management System Over File-Oriented System

Information is kept in relational databases as tables with rows and columns. This configuration makes it straightforward to search for and get particular data from the database. Because the data in each table is connected to one another, it is known as a "relational" database.

Additionally, tables may be linked to one another. In relational databases, tables or files holding data are known as relations (tuples), and they are defined by rows (or records) and columns (or attributes) known as fields. Every table has a key field that serves as the primary identifier of each row (record) and the connections between entries in other tables (or linked).

You can run queries that simultaneously obtain data from several tables thanks to this kind of relational structure. A relational database management system (RDBMS) might also provide the data in a visual format. It might, for instance, offer data in a table resembling a spreadsheet so you can review and even modify particular data points.

Some RDMBS apps let you create forms to make adding, editing, and removing data simpler. The bulk of well-known database management systems fall within the RDBMS group A database management system (DBMS) that facilitates the modelling and generation of data as objects is known as an object-oriented database management system (OODBMS), also abbreviated as ODBMS. Included in this is some type of support for object classes, as well as the ability for subclasses and their objects to inherit class attributes and functions.

object database is Not Only SQL (NoSQL) document database systems. NoSQL document databases offer key-based access to semi-structured data as documents, generally using JavaScript

C. OODBMS Characteristics

Malcolm Atkinson and others defined an OODBMS in their influential work, The Object-Oriented Database Manifesto, as follows:

An object-oriented database system must meet two requirements: it must be a database management system (DBMS) and it must be an object-oriented system, meaning that it must, to the greatest extent feasible, be compatible with the current crop of object-oriented programming languages.

Persistence, secondary storage management, concurrency, recovery, and an ad hoc query facility are the five qualities that correspond to the first criterion.

The second one translates into eight characteristics: extensibility, computational completeness, overriding combined with late binding, inheritance, types or classes, complex objects, object identity, and encapsulation.

D. RDBMS vs. ODBMS

The type of DBMS that is currently used the most frequently is a relational database management system (RDBMS). Most IT professionals have a solid understanding of the relational abstraction of rows and columns accessed using Structured Query Language (SQL).

However, object database systems may be more effective in managing and storing complicated data relationships. Accessing data with several relationships spread across multiple tables in an RDBMS might be more challenging for applications than accessing the same data as an object in an ODBMS.

In addition, a lot of programmers employ object-oriented programming (OOP) languages to create applications, including Java, C++, Python, and others. Conversions between complicated objects and rows from various relational database tables are necessary when using an RDBMS to store and retrieve objects. Tools for object-relational mapping (ORM) can make this process simpler,

The ODBMS has the benefit of eliminating impedance mismatch when designing applications using the OOP methodology; in other words, the software manages and interacts with objects rather than rows of data that must be assembled into an object.

Numerous RDBMS companies now offer object-relational database management systems as part of their product lines (ORDBMS). Of course, adding some object-oriented ideas to relational databases does not give users access to all of an ODBMS's features.

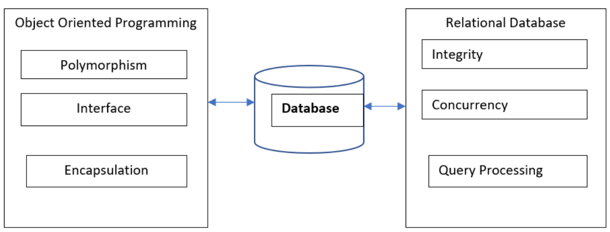

Fig. 2. OODBMS & RDBMS

2. Overview of RDBMS and OODBMS

A. Relational Database Management Systems

This database management system is the most admired and well-liked since it enables a variety of vendor implementations that provide answers to issues using understandable programming languages like Oracle and Visual Basic, while results are supplied for its overall overview. Students and professionals who find this useful for basic reference should only need a minimal amount of training because it can be used with all database systems that operate on any type of computer (Gupta, 2009). As it includes an even more potent retrieval mechanism and update operators, the relational model also offers a distinct physical and logical characteristic of a database, giving users a more intuitive representation of the data. Using short commands, this enables the execution of complex procedures. Most importantly, the RDBMS architecture does give analysts a variety of strong tools that allow them to determine whether a database has inherent problems in its design and when they occur, along with a streamlined interface of the database that is generally consistent. Furthermore, a programmer is not always required for data upkeep in an RDBMS.

B. Object-Oriented Database Management Systems

Since their object classes are compatible with the programming language used for the OODMS models, object-oriented database systems have been crucial to the overall process of application design. The program's object does not need to undergo any special model transformation in order for the database management to understand this. The OODBMS architecture also contains a complex database with a rich data structure that includes class hierarchies, data objects, active data, and methods. These features give the database system flexibility and power while also increasing the complexity and discovery of database information. Object databases also permit their users can arbitrarily build complicated data kinds that are comparable to the programming language concepts, whereas relational databases do not. A high value of removing any impedance mismatch is provided by object databases' additional properties, such as bags, sets, lists, and other containers that are typically appropriate for describing query results that have several objects (Gupta, 2009). Since their extensive data structures extend to the programming language rather than any external service having a constrained and limited interface as seen in relational databases, object databases have a permanent way of carrying data. As they offer massive storage and manipulation of diverse data types like Binary Large Objects (BLOB) and Character Large Objects, object databases also guarantee a strong typing amity for effective communication between applications and the design structure and type of the database system (CLOB).

C. Disadvantages of RDBMS and OODBMS

1) RDBMS shortcomings

One of the most important shortcomings of relational database systems is their spartan design and usage, which can lead to database management systems being poorly developed, designed, and implemented. If the database size is even slightly increased, data corruption, performance degradation, and system slowdown can also occur. RDBMS hardware has a high running cost since its end users must be informed about the implementation's hidden intricacies as well as the specifics of how physical data is stored. Relational database systems are also easier to use and install, but since anybody can make their own applications and databases, problems with data duplication, consistency, and redundancy may occur in an organizational setting (Ward & Dafoulas, 2006). The information integration that is much needed will also be hampered by this. Furthermore, compared to object database systems, relational database systems are unable to handle more complicated and advanced applications at a faster rate. In contrast to object database systems, which do not require such conditions, this type of system does not enable inheritance, does not have set-valued attributes, and heavily relies on programmers to specify the data access language and the host language in order to resolve impedance mismatches.

2) OODBMS drawbacks

This database system's lack of a clear description of what it is due to the word itself being applied to a range of prototypes and products that are all very different from one another is one of its key drawbacks. When comparing migration of databases to relational databases, which do not impose on periodic change, it is similarly challenging to manage because the definition of objects changes periodically. It is also not commonly utilised because it is important in circumstances when complicated administration of data items is required. In other words, it is intended only for a select few uses (Ward &

Dafoulas, 2006). Due to the high processing requirements, it also has a high maintenance cost.

3) *Security issues*

3. Advantages and Disadvantages of Hybrid Systems Over RDBMS and OODBMS

A new generation of a hybrid system was introduced as the industry gained momentum with the introduction of commercialized database systems because it embodies the concept of both relational and object-oriented database systems because it stores data in the OO formation or relational formation, allowing programmers to access the information in their preferred manner. Since both the RDBMS and OODBMS integrate significant signals of strengths and shortcomings, this hybrid system is either known as an object-relational database system (ORDBMS) or as well as an enhanced relational database system (ERDBMS). This configuration of the system combines data from the RDBMS and OODBMS in order to fully utilise sophisticated applications with different design structures from the two traditional database systems (Baba, Jain, & Howlett, 2001). The only setback is that since it simplifies data relationships it has a potential chance of lowering data efficiency.

4. Relational Database Management Systems in Business

A manager of a small business has historically been able to maintain critical data using a manual file system, and in some cases, they still can. The manual system did a good job of serving as a data repository as long as the data gathering was modest and management of the company didn't need many reports. However, maintaining data in a manual file system became more challenging as firms grew and reporting requirements grew more sophisticated. The likelihood that data would ever produce usable information decreased as discovering and using the data in the expanding collections of file folders became such a laborious effort. As a result, databases were created as a remedy for file system problems.

The database consists of logically related data saved in a single repository, which is typically stored in the computer, as opposed to the file system, which has numerous distinct and unrelated files. The majority of the file system's difficulties with data inconsistency, data anomalies, data dependency, and structural dependency can be solved by the database's Database Management System (DBMS). The structure of the data and the connections between them, or database models, can be used to categorise DBMS. Database models include the relational database model, network database model, and hierarchical database model, among others. Tables are used to hold data in relational databases, making data storage and manipulation more effective and nonredundant (Rob & Coronel, 2000). The programme that aids users in relational database design, management, and implementation is known as a Relational Database Management System (RDBMS). Therefore, the relational database model and its use in business are the main topics of this essay. The article covers the basic steps in relational database design, the structure of a relational database model, examples of RDBMS and their applications, the significance of RDBMS in modern company, the duties of a database administrator, and the value of RDBMS in today's business.

Basic Relational Database Model Structure The relational database paradigm, first created by Codd in 1970, is implemented through an extremely complicated RDBMS. RDBMS are widely used in government and healthcare organizations today because they can handle a wide range of tasks, including data dictionaries management, storage management, data transformation and presentation, security management, data integrity management, backup and recovery, and data transformation and presentation. As a result, using and understanding the relational database construct is made simpler. In addition, because the RDBMS handles the intricate physical elements, users or designers can work on a human logical scenario.

Users view the relational database as a collection of tables or relations where data is kept. There are several rows and column intersections in every table. A shared entity characteristic links tables to one another. Thus, the relational model offers a minimal amount of controlled redundancy to get rid of the vast majority of redundancies typically present in file systems. 2000 (Rob & Coronel).

Examples of RDBMS and their use:

The most popular RDBMS on the market right now are Microsoft Access and Oracle, among others. Both databases are relational DBMSs, which offer effective and non-redundant data handling and storage. Thus, the characteristics of these database applications as they are used in organizations are as follows: data is entered and stored once in tables and does not need to be repeated; sharing of data from a remote storage and through the network is allowed; only related data are displayed so as to eliminate irrelevance and ambiguity; and they limit the volume of data in a table to only related fields so as to facilitate quick and easy data search. Since Access is a part of the hugely popular Microsoft Office software suite, data can be shared with other office programmes like Microsoft Excel, Microsoft Word, and Microsoft PowerPoint. Management of organizations like the medical industry and government agencies view Access as a feature-rich programme that can handle any database-related task, including storage and even database tools. The company can also easily publish its information to the internet via the World Wide Web thanks to Access. This boosts productivity for the business by giving its clients direct information under the guiding principle of "offering the best of both worlds" (Laudon & Laudon, 2006).

However, because of its ability to manage data, Oracle is a database application that is highly valued by government organisations like the Bureau of Census. As an Oracle version, Oracle Database 10g is far more effective and, more crucially, its security system guarantees that an agency's sensitive data is safeguarded and maintained effectively. The Oracle application can be used by other programming languages like Java that rely on the Application Programming Interface to call stored procedures and functions as well as store and execute stored procedures and functions automatically (API). Additionally, Oracle stores data both physically in the form of data files and logically in the form of tablespaces, making it easier for database administrators to administer databases (Oracle, 2010).

RDBMSs are typically used by enterprises because they offer the following features:

1. *Data storage management:* Without knowing any programming lingo, users can store and retrieve data from the data repository using Oracle and Microsoft Access. For instance, a patient's admittance number can be used to quickly obtain their records. RDBMSs additionally offer data storage and related data entry forms;
2. *Data conversion and presentation:* Database applications convert the system's actual physical data structure into a more intuitive conceptual format;
3. *Security administration:* The database apps make sure that data and security policies are in sync. Discretionary access control, for instance, is a feature of the Oracle server that guarantees that access to information is controlled by privileges;
4. *Transaction support:* Transaction control is crucial because the majority of firms, like the financial sector, are concerned with processing a lot of data. Therefore, database applications guarantee that the database is correctly updated when several users access it;
5. Data integrity and safety are ensured through backup and recovery management;
6. Data integrity management, which is supported and regulated by RDBMS;
7. They include specialized communication protocols that enable the database to accept user queries (Stair & Reynolds, 2006).

5. Importance of RDBMS in Businesses Today

A. Setting Relationships between Entities

The concept of relationships between items or entities serves as the foundation for the RDBMS. An association of entities based on their interactions is called a relationship. In order to model a relational database model, the following terminology are used: Entity - an individual, location, object, or event from which data is to be gathered; Attributes serve to characterize the object they are connected with. A field called the primary key identifies each row in a row individually. and A foreign key is a key that a primary key value in another table must match for there to be a relationship or link between the two tables. In essence, one-to-one, one-to-many, and many-to-many connections can be formed between entities using an RDBMS.

1) One-to-One Relationship

This kind of relationship emerges when just one instance of the second entity appears for every one of the first. As a result, it follows that each record in table Y must only have one record in table X that matches it. Likewise, each record in table X must only have one record that matches it in table Y. When enforcing corporate standards, this kind of interaction is crucial, especially when connecting two tables. One-to-one relationships can exist in a single table or could be used to divide one table with multiple fields into records that are entirely dependent on the primary key (Harkins, 2003).

A corporation might be an example of a one-to-one relationship if there is only one manager allowed to lead each department, which results in a one-to-one relationship. The terms manager and department can be separated into two different tables, MANAGERS and DEPARTMENTS. A primary key (ManagerID) with a "unique" property that uniquely identifies each manager may be present in the MANAGERS table. On the other hand, the DEPARTMENTS table would have other entities like the department name and employees in addition to the ManagerID, a special field. The primary key's use establishes the association. As a result, the two tables have a one-to-one relationship when they are connected. Here is the common shorthand representation:

MANAGERS (Manager_ID, Name, Age, Address, Salary)

Foreign Key: Manager_ID to DEPARTMENTS Table

DEPARTMENTS (Manager_ID, Department_Name, Employees)

Foreign Key: Manager_ID to MANAGERS Table

2) One-to-Many Relationship

Assuming that each instance of the second entity can only be associated with one instance of the first entity, this form of relationship is frequently employed. It presupposes that one instance of the first entity can be associated with multiple instances of the second entity. In essence, a record in table X may match numerous records in table Y, for example. If only one of the related fields is a unique field, then this kind of relationship can be realised (primary key).

In a hospital, there may be a one-to-many relationship where one nurse cares for numerous patients while another cares for just one. Two tables, the PATIENTS table and the NURSES table, can be used to describe this type of one-to-many interaction. Patient ID, the primary key for the PATIENTS database, and Nurse Number, effectively the key attribute for the NURSES table, are examples of a foreign key. Nurse Number and Patient ID make up the composite key that is contained in the NURSES table. The main key is comprised of these keys. The foreign key to the PATIENTS table in this case is Patient ID. Therefore, a one-to-many relationship is created when the two tables are linked.

Here is the common shorthand representation:

PATIENTS (Patient_ID, Name, Address, Ward Name, State, Zip, Nurse_Number)

Foreign Key: Nurse_Number to NURSES Table

NURSES (Nurse_Number, Patient_ID, Name)

Foreign Key: Patient_ID to PATIENTS Table

3) Many-to-Many Relationship

One instance of the first entity can be linked to numerous instances of the second entity, and vice versa, with one instance of the second entity linking numerous instances of the first entity. To put it another way, a record in database X may contain numerous matching records in another table, let's say table Y, and vice versa. To enforce this relationship, a junction table or link table that contains the primary keys from both tables can be utilised. Consequently, the relationship becomes two or more one-to-many relationships (Begg & Connoly, 2003).

A school database is an example of a many-to-many relationship since it allows students to enrol in multiple courses at the same time and multiple students to enrol in the same course. Three tables, STUDENTS, COURSES, and ENROLLMENT, which serve as a junction table linking the other tables, can be defined in this situation. The Student ID primary key of the STUDENTS database may also serve as a foreign key to the ENROLLMENT table. The Course ID column of the COURSES database would serve as the key attribute and serve as a foreign key to the ENROLLMENT table. Enrollment ID is a unique field in the junction table ENROLLMENT, and Student ID and Course ID serve as foreign keys to the table.Here is the common shorthand representation:

STUDENTS (Student_ID, Name, Age, Enrollement_Date)
Foreign Key: Student_ID to ENROLLMENT table
ENROLLMENT (Enrolement_ID, Student_ID, Enrollment_Date, Course_ID)
Foreign Key: Student_ID to STUDENTS table
Foreign Key: Course_ID to COURSES table
COURSES (Course_ID, Name, Teacher)
Foreign Key: Course_ID to ENROLLMENT table

Therefore, using the concept of relationships enable managers of various businesses to make decisions that are not biased because most of the businesses entities and processes are related in some way, and that is why the RDMS is useful in record searching.

B. Improving Entity Integrity

In various ways, RDBMS is used to improve the integrity of data. Every table in a relational database must have a primary key in order to maintain entity integrity, and duplicate entries inside a table are never permitted. A database designer chooses the appropriate primary key that matches the requirements of the database with information search, updating, and deletion of the stored data to ensure that the data is valid and accurate (Harkins, 2003). Tables like the WARDS table and PATIENTS table, for instance, are included while developing a hospital database.

Referential integrity is another sort of data integrity. When records are added or deleted, these rules must be observed in order to maintain the stated relationships between tables. An arrangement of a primary key with a foreign key is often used to enforce referential integrity. These two keys need to be present in order to connect one table to another. The EMPLOYEE table and the DEPARTMENT table in a company database serve as an example of referential integrity. Employee number serves as a primary key and Department number serves as a foreign key to connect the EMPLOYEE table to the DEPARTMENT table, guaranteeing consistency in relationships. The RDBMS often offers referential integrity enforcement (Rob & Coronel, 2000).

Because they guarantee that data is accurate and consistent and as a result allows for the establishment of relationships, entity integrity and referential integrity are crucial to relational databases. Second, because records may only be entered once, they lessen data redundancy. Thirdly, using primary keys to search records makes the process considerably quicker.

6. Conclusion

Object-oriented database systems and the partially employed hybrid systems are currently being developed and implemented by a number of businesses despite the fact that RDBMS systems still control the majority of the market. This is because these new-generation products almost perfectly "fit" the needs of the contemporary market and DBMS. Due to the impressive expansion of OODBMS and applications, even more effective and potent tools for data analysis are now required. The best recommendation is to adopt a hybrid database system since it combines the benefits of both RDBMS and OODBMS technologies, has fewer problems, and allows all users to benefit from its OO architecture and capabilities for complete data integration.

References

[1] Baba, N., Jain, L. C. & Howlett, R. J. (2001). Knowledge-based intelligent information Engineering systems & allied technologies: KES'2001, Volume 5. Amsterdam, Netherlands: IOS Press.
[2] Gupta, D. (2009). Database Management System Oracle SQL and Pl/SQL. New Delhi: PHI Learning Pvt. Ltd.
[3] Perry, J.T. & Newmark, R. (2010). Building Accounting Systems Using Access 2010. Mason, OH: Cengage Learning.
[4] Ward, P. & Dafoulas, G. (2006). Database Management Systems. Cengage Learning EMEA.
[5] Begg, C., & Connoly, T. (2003). Database Systems. (3rd Ed.). Delhi: Pearson education Limited.
[6] Harkins S. (2003). Relational Databases: Defining Relationships between Database Tables. Tech. Republic.
[7] Laudon, K. C., & Laudon, J. P. (2006). Management information Systems: Managing the Digital Firm. (9th Ed). Upper Saddle River, NJ: Pearson Prentice Hall.
[8] Rob, P. & Coronel, C. (2000). Database Systems: Design, Implementation, and Management. (4th Ed.). Cambridge: Thomson Course Technology.
[9] Stair, R. M., & Reynolds, G. W. (2006). Fundamentals of information systems. (3rd Ed.). Boston: Thompson Course Technology.