

A COMPARATIVE STUDY BETWEEN THE PERFORMANCE OF RELATIONAL & OBJECT ORIENTED DATABASE IN DATA WAREHOUSING

Dr. (Mrs Pushpa Suri)¹ and Mrs Meenakshi Sharma²

¹ Associate professor, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India
pushpa.suri@yahoo.com

² Assistant professor, Department of Computer Science and Engineering, Haryana College of Technology and Management Kaithal, India
minny_kaushik@yahoo.com

ABSTRACT

A data warehouse is a repository of subjectively selected and adapted operational data which can successfully answer any ad hoc, statistical, complex or analytical queries. Data warehousing technology is becoming essential for effective business intelligence, business strategy formulation and implementation in a globally competitive environment where in larger and larger amounts of data are required to be processed faster and faster for comprehension of its real meaning and impact. A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process". Data that gives information about a particular subject instead of about a company's ongoing operations. It is integrated as data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole. Data warehouse system is time variant as all data in the data warehouse is identified with a particular time period This paper discusses in detail two existing database models used for data warehousing, the relational database and object oriented database and does an analysis of the benefits of Object Oriented databases over the conventional relational database in Data warehousing.

Keywords: Data warehousing, RDBMS, Object oriented data warehousing

BACKGROUND OF THE STUDY

1. INTRODUCTION TO RELATIONAL DATABASE

According to **Inman W (1993)[1]** a relational database consists of a collection of tables each of which is assigned a unique name. The relational model is the primary data model for commercial data processing applications. It is being used in a number of applications outside the domain of the traditional data processing. Relational databases allow us the users to store and retrieve data in a tabular form. **Jae Jin Koh (2007)[11]** says that a relational database is a group of tables, also called relations in the database community. A relational database is a database that conforms to

the relational model and could also be defined as a set of relations or a database built in an RDMS. The below figure shows the relational database:

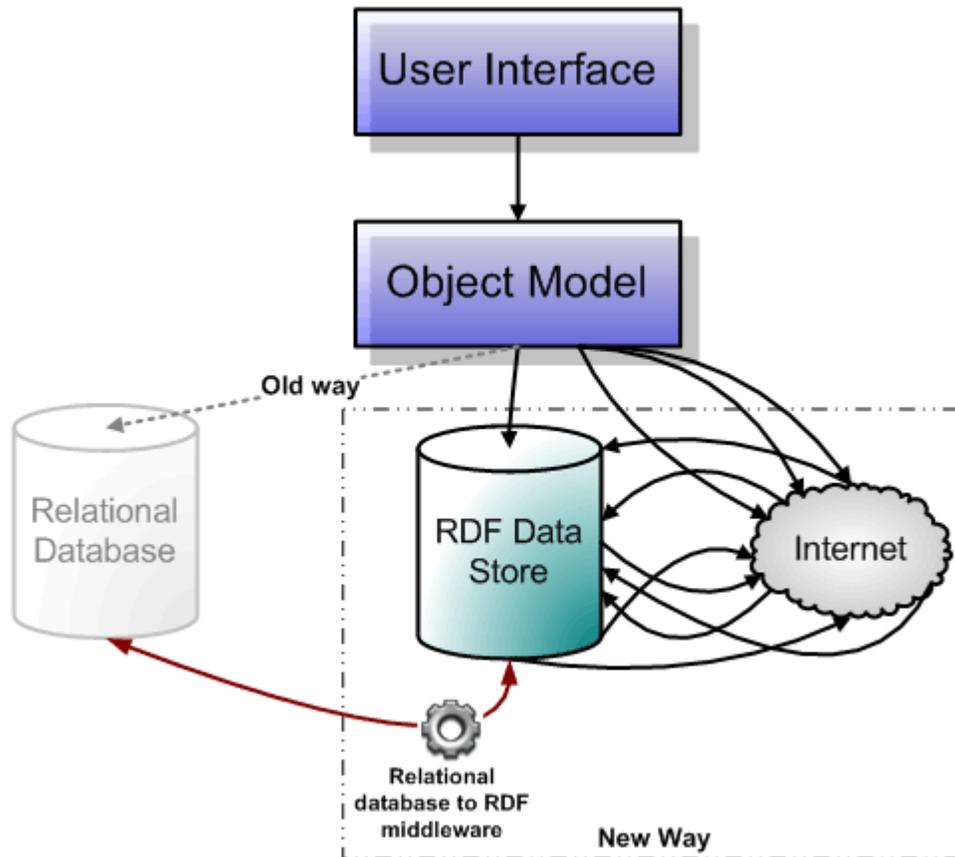


Figure 1: Relational database

Source: Mary E S (1992), ODBMS vs. Relational Object-Oriented Programming, SAGE, London

A relational database management system is a system that manages data using the relational model. Most current RDMS for example My SQL, Oracle, Microsoft SQL Server, and PostgreSQL are more accurately called SQL database management products. A database based on the relational model is developed by E F Codd. A relational database allows the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data and the relations between them are organized in tables. A table is a collection of records and each record in a table contains the same fields. In common practice relational databases have become the most significant and central platforms for data management and query solving in the current computing field. O matter which software application o online websites relational databases are employed to control data manipulation. The accuracy and efficiency of query solving is an issue of concern and has been improved among various database systems. However with increasingly complex requirements for queries from the customers a problem is triggered ad combinational

queries cannot be efficiently solved by the conventional relational database system. Current RDMS need much more time and effort to evaluate complex combinational queries. The definition of a combinational query is a request for tuples from multiple relations that satisfy a conjunction of constraints on tuple attribute values.

Darter C J (1985) [2] says that the relational model represents the database as a collection of relations. Each relation corresponds to a table of values or to some extent a flat file of records. The terms that were previously used to describe the database properties and functions lacked the precision necessary for the formal theory that Codd was proposing. So a new set of terminology had to be evolved. The following tables give a list of the relational terms and their corresponding informal equivalents.

<i>Relational terms</i>	<i>Previously used terms</i>
Relation	Table
Tuple	Row, Record
Cardinality	Number of Rows
Attribute	Column, Field
Degree	Number of Columns
Primary Key	Unique Identifier
Domain	Set of Legal Values

The relational database management systems are based on the relational model. In the relational model the data and the relationships among data are represented by a collection of tables. A row in a table represents a relationship among a set of values. There is a similarity between the concept of table and the mathematical concept of relation.

2. WHAT IS DATA WAREHOUSING?

Mary S L (1995)[3] says that a data warehouse is a repository of subjectively selected and adapted operational data which can successfully answer any ad hoc, statistical, complex or analytical queries. It is situated at the centre of a decision support system of an organization and contains historical integrated data., both summarized and detailed information common to the entire organization. Data warehousing technology is becoming essential for effective business intelligence, business strategy formulation and implementation in a globally competitive environment where in larger and larger amounts of data are required to be processed faster and faster for comprehension of its real meaning and impact. Data warehousing enables easy organization and maintenance of large data in addition to fast retrieval and analysis in the manner and depth required from time to time. The below figure shows the general diagram of data warehouse:

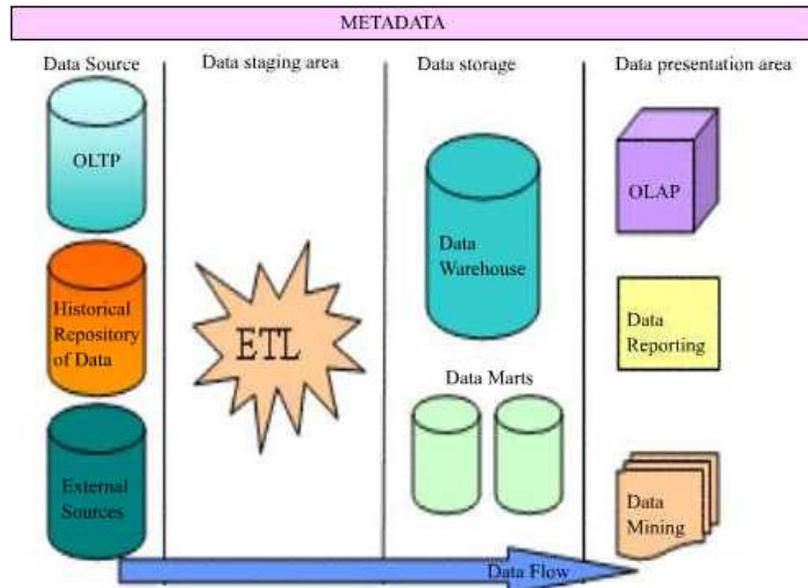


Figure 2: Data Warehouse Source: Website, Argowiki.com

A data warehouse is a powerful database model that significantly enhances the user's ability to quickly analyze large, multidimensional data sets. It cleanses and organizes data to allow users to make business decisions based on facts. Hence the data in the data warehouse must have strong analytical characteristics. Creating data to be analytical requires that it be subject oriented, integrated, time referenced and on volatile.

1. Subject Oriented Data:

According to **Arthur M K (1993)[4]** data warehouse groups data by subject rather than activity. In contrast transactional systems are organized around activities such as payroll processing, shipping products, loan processing and the like. In data warehouse environment information used for analysis is organized around subjects such as employees, account sales, products and so on. This subject specific design helps in reducing the query response time by searching through very few records to get an answer to the user's question.

2. Integrated Data:

Integrated data refers to de-duplicating information and merging it from many sources into one consistent location. Much of the transformation and loading work that goes into the data warehouse is centered on integrating data and standardizing it.

3. Time referenced Data:

The most important and most scrutinized characteristic of the analytical data is its prior state of being. In the words time referenced data essentially refers to its time valued characteristic. Time

referenced data when analyzed can also help in spotting the hidden trends between different associative data elements which may not be obvious to the naked eye. This exploration activity is termed data mining.

4. Non Volatile Data:

Since the information in a data warehouse is heavily queried against time it is extremely important to preserve it pertaining to each and every business event of the company. The non volatility data, characteristic of data warehouse enables users to dig deep into history and arrive at specific business decisions based on facts.

3. LIMITATIONS OF USING RELATIONAL DATABASES IN DATA WAREHOUSING

The limitations of relational database in data warehousing are:

i. Processing Efficiencies

Because large scale databases may be composed of many interrelated tables, the overall design may be complex and therefore have slower reach and access times in comparison to the hierarchical and network models. The slower search and the access time may result in processing efficiencies which lead to an initial lack of acceptance of the relational model.

ii. Hardware Overheads:

The relational database management system needs comparatively powerful hardware as it hides the implementation complexities and the physical data storage details from the users. For these relational database systems needs more powerful computers and data storage devices. So the relational database management systems need more powerful computers to run smoothly. But as the processing power of modern computers is increasing at an exponential rate and in today's scenario the need for more processing power is not a big issue.

iii. Ease of design can result in bad design:

Park M (1996)[6] says that the relational database is an easy to use design and use system. The users need not know the complex details of physical data storage. They need not know how the data is actually stored to access it. This ease of design and use can lead to the development and implementation of very poorly designed database management systems. Since the database is efficient, these design efficiencies will not come into the picture when the database is designed and when there is only a small amount of data. As the size of the database increases, the poorly designed database will slow the system down, and will result in performance degradation and data corruption.

iv. Information Island phenomenon:

As the relational database systems are easy to use and implement, people or departments may create their own database and applications. This situation might hinder information integration that is necessary for the smooth and efficient functioning of the organization. Problems like data inconsistency, data duplication and data redundancy may also crop up.

v. Lots of Information

Advancement in the information complexity can cause another drawback to the relational databases. Relational databases are made for collecting data by similar characteristics. Complex numbers, images, multimedia and design products oppose easy categorization which leads the way for a new database type called object-relational database management systems. These systems have the ability to be scalable and are designed to handle more complex applications.

vi. Structured Limits:

Some relational databases have limits in the field lengths. When a database is designed the amount of data must be specified which can fit into the field. Search queries or some names are shorter than the original and this can lead to loss of data.

4. OBJECT ORIENTED DATA WAREHOUSING - A NEW APPROACH

According to Molina G H (1995) [7] the object oriented data warehouse architecture are explained based on the proposed logical architecture. The differences between these architectures are the new metadata layer and “the object-orientation” approach. With the approach of object-oriented most layers in this architecture has the “Data Store” layer which consists of various types of objects which performs the underlying functions of each component. In this architecture, the flow of data is same to other architectures of data warehouse where the data can be gathered from diverse operational database systems, that are integrated, aggregated and summarized in the data warehouse, and it uses read-only data to support the analysis of complex data types. The object oriented data warehouse architecture consists of the following components. They are:

Application interface layer

In the application interface layer, the objects of this component hide the complex processes of data from the user of data warehouse. The objects of this component are categorized into different groups which serves various services. Based on the application layer functionalities, each service gives response to the coming requests of the data analyzers, third party applications or other users of the data warehouse system. For data warehouse usage the main functions of the types of object of this layer is to receive the queries of user, and process the queries and then send the final request to the component of Data Warehouse Management. After that they obtain the results of queries from the core layer. In this architecture, a query is not executed directly at this layer. For

administering the data warehouse operations the objects of this layer will provide functions to manage the services of user, control the updates, and finally maintain the data processes of the warehouse. This means, that the services of new user can be added in this layer to support the new requirements for user if needed.

Data Acquisition

The component of data acquisition is a tool which constructs the data warehouse and the data engine. The objects of data acquisition will transfer, extract and transform the data from various legacy operational data stores to the data warehouse of the object oriented data warehouse. The functions of this component are categorized into the sub-function levels that can be performed by the pattern types of objects.

Managing the Data Warehouse

Roussopoulos N (1997) [8] says that this component layer accesses the data directly from the data warehouse from the object oriented data warehouse. It provides the services, which separates the object oriented database and the application interface layer. In this component, various methods are applied to access the stored data in the object oriented data warehouse. Further the access methods of database can be added or updated to improve the data warehouse performance. This layer can be classified into 2 individual components which allow to distinguish clearly between the data input processing and the read-only data processing in data warehouse. The functions of this component are to read the necessary data, and to create new views of materials based on the data.

Metadata

In an object oriented data warehouse the metadata is defined for the metadata behavior objects which depend on their individual roles. For example, metadata can count its available frequency itself, by making the query usages statistics and so on. This means that many queries about the operations of warehouse can be answered easily by querying the metadata directly.

Data store

The data stored in the object oriented data warehouse differs from the data warehouse in the relational environment. Depending on the data types and requirements the designers of object oriented data warehouse can decide to model it as object hierarchy or as “cube” like structure just as Multidimensional OLAP, or like Object-Oriented OLAP. For example, in object oriented data warehouse simple data can be modeled in the structure of multidimensional data that looks like what must be done in the relational database systems. Otherwise, the user defined and the complex data can be modeled in the structures of object hierarchy that was suggested for object oriented database management systems. Further the objects of any layer specifically the objects of metadata can be modeled in the object oriented database.

5. THE ADVANTAGES OF OO DATABASE OVER RELATIONAL DATABASE:

(i) Relational Databases

According to **Swift R (1996) [9]** relational databases have dominated the database world for decades and they are particularly well suited for business data which usually lend themselves to structuring in the form of tables. Relational databases can be accessed by means of object oriented programming languages. However that does not turn a relational database into an object oriented one. In relational database model data are logically organized in two dimensional tables. Each individual fact or type of information is stored in its own table. The relational database model was developed using a branch of mathematics called set theory. In set theory a two dimensional collection of information is called a relation. A relational database management system allows users to query the tables to obtain information from one or more tables in a very flexible way. The relational database is attractive from a user's standpoint because end users often think of the data they need as a table. The capability of a relational database management system to handle complex queries is important. Although the relational database is considered to be a dramatic improvement over the network and relational models, it does not have two disadvantages. First a relational database requires much more computer memory and processing time than the earlier models. Increases in computer processing speeds as well as a steady decrease in hardware costs have reduced the impact of this first disadvantage. The second disadvantage is that the relational database allows only text and numerical information to be stored in the database. It did not allow the inclusion of complex object types such as graphics, video, audio, or geographic information. The desire to include these complex objects in databases led to the development of object oriented databases.

(ii) Object Oriented Database

Marlin E S (1992) [10] says that object oriented databases can store free standing objects. In recent years there has been a trend in the direction of object oriented programming languages, object oriented databases have found only a small market niche. Object oriented database systems enable direct access to objects defined in the programming language in question and the storage of such objects in the database without conversion. It is precisely this that is not possible with relational database systems, in which everything must be structured in tables. Both simple and complex objects can be stored in an object oriented database model. In object oriented database other types of data can be stored. Object oriented database includes abstract data types that allow the users to define the characteristics of the data to be stored when developing an application. This overcomes the limitations of relational databases. Relational databases limit the types of data that can be stored in table columns. Instead of tables an object oriented database stores the data in objects.

An object can store attributes and instructions for actions that can be performed on the object or its attributes. These instructions are called encapsulated methods. Objects can be placed in a hierarchy attributes from objects higher in the hierarchy. Many researchers have argued that the object oriented database are superior to relational database most organizations still use relational

database. The main advantage of an object oriented database is to store complex data types. In recent years object oriented relational database have been developed Ambhore, P.B. (2007) [5]. This combined structure includes a relational database framework with the capability to store complex data types. Although developers continue to refine object oriented database most companies seem to be satisfied with relational or object oriented database at this time. The advantages of the object oriented database are:

- **Capable to handle many different types of data types:** An object oriented database can store any type of data including text, numbers, pictures, video and voice.
- **Combination of object oriented programming and database technology:** The object oriented database technology combines object oriented programming with the database technology to provide an integrated application development system. There are many advantages of including the definition of operations with the definition of data. First the operations which are defined are applied and are not depending on the specific database application running at the moment. Second the data types can be extended to support complex data.
- **Improves productivity:** Inheritance allows programmers to develop solutions to complex problems by defining the new objects in terms of defined objects in the past. Polymorphism and dynamic binding allows programmers to define operations for one object and then to share the specification of the operation with other object. These objects can further extend this operation to provide behaviors that are unique to those objects.
- **Data access:** Object oriented databases represent relationships explicitly supporting both navigational and associative access to information.

6. COMPARISONS OF OODBMS TO RDBMS

There are concepts in the relational database model that are similar to those in the object database model. A relation or table in a relational database can be considered to be analogous to a class in an object database. A tuple is similar to an instance of a class but is different in that it has attributes but no behaviors. A column in a tuple is similar to a class attribute except that a column can hold only primitive data types while a class attribute can hold data of any type. Finally classes have methods which are computationally complete (meaning that general purpose control and computational structures are provided [McF 99]) while relational databases typically do not have computationally complete programming capabilities although some stored procedure languages come close. Below is a list of advantages and disadvantages of using an OODBMS over an RDBMS with an object oriented programming language.

Advantages

1. **Composite Objects and Relationships:** Objects in an OODBMS can store an arbitrary number of atomic types as well as other objects. It is thus possible to have a large class which holds many medium sized classes which themselves hold many smaller classes, ad infinitum. In a relational database this has to be done either by having one huge table with lots of null fields or via a number of smaller, normalized tables which are linked via foreign keys. Having lots of smaller tables is still a problem since a join has to be performed every time one wants to query data based on the "Has-a" relationship between the entities. Also an object is a better model of the real world entity than the relational tuples with regards to complex objects. The fact that an OODBMS is better suited to handling complex, interrelated data than an RDBMS means that an OODBMS can outperform an RDBMS by ten to a thousand times depending on the complexity of the data being handled.
2. **Class Hierarchy:** Data in the real world is usually has a hierarchical characteristic. The ever popular Employee example used in most RDBMS texts is easier to describe in an OODBMS than in an RDBMS. An Employee can be a Manager or not, this is usually done in an RDBMS by having a type identifier field or creating another table which uses foreign keys to indicate the relationship between Managers and Employees. In an OODBMS, the Employee class is simply a parent class of the Manager class.
3. **Circumventing the Need for a Query Language:** A query language is not necessary for accessing data from an OODBMS unlike an RDBMS since interaction with the database is done by transparently accessing objects. It is still possible to use queries in an OODBMS however.
4. **No Impedence Mismatch:** In a typical application that uses an object oriented programming language and an RDBMS, a significant amount of time is usually spent mapping tables to objects and back. There are also various problems that can occur when the atomic types in the database do not map cleanly to the atomic types in the programming language and vice versa. This "impedance mismatch" is completely avoided when using an OODBMS.
5. **No Primary Keys:** The user of an RDBMS has to worry about uniquely identifying tuples by their values and making sure that no two tuples have the same primary key values to avoid error conditions. In an OODBMS, the unique identification of objects is done behind the scenes via OIDs and is completely invisible to the user. Thus there is no limitation on the values that can be stored in an object.
6. **One Data Model:** A data model typically should model entities and their relationships, constraints and operations that change the states of the data in the system. With an RDBMS it is not possible to model the dynamic operations or rules that change the state of the data in the system because this is beyond the scope of the database. Thus applications that use RDBMS systems usually have an Entity Relationship diagram to model the static parts of the system and a separate model for the operations and behaviors of entities in the application.

With an OODBMS there is no disconnecting between the database model and the application model because the entities are just other objects in the system. An entire application can thus be comprehensively modeled in one UML diagram.

7. THE FUTURE

Michael S (1991) [12] concludes that in this study the migration from the relational database management system to object oriented database management system are focused. It is too much to ask for the framework of an application to support natural and deft objects manipulation in both the on line (object oriented database management system) and off line (relational database management system) forms. In any case, the issues of data representation and the subject of research in the community of academic database are not the crucial problems. Instead, the core issues lie in the areas that recognize the most persistent data where the transaction semantics, object identity and the naming of object are denoted. The object oriented database management system provides fast, direct, and application-pertinent access of object and the Relational database provides a generalized evolution with clear representation. The solutions of long-term may be a hybrid system, in which the object oriented database management system manages the data that can be erased to the Relational database when the evolution of data is required.

8. CONCLUSION

In this paper we have discussed the concept of object oriented data warehousing. For complex data the object oriented database management system is faster and better than relational database management system. For object oriented database management system, the goal is to direct the persistent objects manipulations by the application software. The inseparable practices are the operational burdens and the semantic burdens who attend the direct manipulation. Data warehouses contain consolidated data from many databases and other external data sources, spanning long time periods, and augmented with summary information. There are many new challenges in designing, creating and maintaining large data warehouses.[14],[16],[3],[4] There are several new promising research issues some of which are related to problems that the database community has worked on for several years, but others are only just beginning to be addressed. For modern (i.e. object–relational or object–oriented) data warehousing systems the mechanisms of increasing their efficiency have to be more sophisticated, as data to be processed are of arbitrary complex structure and behavior. In the future we will try to enhance our research for more friendly environmental domains so as to reach the more flexible object oriented data warehousing system.

REFERENCES

1. Inman W (1993), *Data Warehouse Building*, John Wiley & Sons, London.
2. Darter C J (1995), *Introduction to Database System*, Addison Wesley, New York.
3. Mary S L (1995), *Object oriented Databases: An Introduction*, Addison-Wesley, New York.

4. Arthur M K (1993), Bridging relational databases and object-oriented programming, Conf. Management of Data, Washington.
5. Ambhore, P.B. (August 20-22, 2007), A Implementation of Object Oriented Database Security, *in* 5th ACIS International Conference on Software Engineering Research (2007), Institute of Electrical and Electronic Engineers, IEEE Computer Society.
6. Park M (1996), Data Warehouse Designing on Relational Database Systems, Informix Co., Stanford.
7. Molina G H (1995), Maintenance in Data Warehousing Environment, San Jose Co., California.
8. Roussopoulos N (1997), Data Warehouses and Materialized Views, Leander Press, Greece.
9. Swift R (1996), Building Advanced Data Warehouse, NCR Corporation, California.
10. Marlin E S (1992), ODBMS vs. Relational Object-Oriented Programming, SAGE, London.
11. Jae Jin Koh,(3-6 October, 2007), Relational database schema integration by overlay and redundancy elimination methods, in International Forum on Strategic Technology(2007), Institute of Electrical and Electronic Engineers, IEEE Computer Society.
12. Michael S (1991), The next generation DBMS, Pearson Education, New York.
13. Bertino E.(1991): Method precomputation in object-oriented databases. Proceedings of ACM-SIGOIS and IEEE-TC-OA International Conference on Organizational Computing Systems.
14. Eder J., Frank H., Liebhart W. (1994): Optimization of Object-Oriented Queries by Inverse Methods. Proceedings of East/West Database Workshop, Austria.
15. E. Bertino et al(2004): Object Oriented Query Languages: The Notion and Issues, IEEE TKDE, vol4, No.3, June.
16. S. Chaudhuri and U. Dyal, "An overview of Data warehousing and OLAP Technology", ACM SIGMOD record, vol 21, no.1.
17. W.C. Chen, T.P Hong and W.Y. Lin, "Object- Oriented data warehousing and its maintenance technology", technical note, 2000.
18. W.C. Chen, T.P Hong and W.Y. Lin, "View maintenance in an object oriented data warehousing", Proceeding of the Fourth international conference on computer science and informatics, North Carolina, USA, pp 353-356, 1998.
19. N.Hyun, "Efficient self maintenance of materialized views", Technical note, 1996.
20. W.C chen, "Object Oriented data warehousing and its maintenance technologies, Master Dissertation, I-Shou University, Taiwan, R.O.C April, 1999.