CHAPTER - 2
LITERATURE REVIEW

2.1 LITERATURE SURVEY

The difficulty and failure implementation of data warehouse technology were stated in the literature. But the research on performance strategy for data warehouse implementation is rare and split. Unfortunately the majority of the available research focused largely on technological and educational aspects, which represent the operational level in the organization. Past studies on data warehousing explained data extraction, cleansing and loading process, by investigating a single or a couple of issue. This obviously has led to lack of discover the impact of these scope, which represent the managerial and strategic levels in the organization.

Data warehousing is a main part of research and development, relatively few research have been conducted to review data warehousing practices in particular. The literature is full of researchers and developers accounts of data warehousing projects that have been successful or unsuccessful and the possible reasons for these outcomes. Some attempts have been made to review their claim. After review and investigated data warehousing implementation at selected companies. It would be helpful to test the claims, made by researchers and findings from case studies in survey. The handfuls of data warehousing surveys that have been published to date are briefly reviewed in the next paragraphs.

S. Chakravarthy and D. Lomet, editors. *Special Issue on Active Databases,* IEEE Data Engineering Bullet in 15[4), December 1992 a lot of changes needed to be carried out to keep the performance graph in a positive direction. Some training courses needed to be introduced, some changed are needed for Help counter, some indexes become obsolete and others need to be created, some aggregates are no longer referenced and others need to be evaluated, and the limits on parallel processing must be assessed and adjusted to fit the current demand. These and other tuning tasks should be carried out periodically Help and support to keep data warehouse performance smooth and constant. Data warehousing is becoming an increasingly important technology for information integration and data analysis [105].

Fred R. McFadden in their paper entitled "Data Warehouse for EIS: Some Issues and Impacts" at 29th Annual Hawaii International Conference on System Sciences - 1996.This paper proposes a research study and includes a data model that relates user satisfaction and development characteristic to organizational factors, the warehouse infrastructure, and management support. Many research opportunities exist in data warehouse and in EIS and DSS[210].

In one of the 1997 paper on "facilitating corporate knowledge: building the data warehouse", in Information management & computer security by Hurley and Harris describe by survey in KPMG management consulting and the Nolan Norton institute. The main goal of this survey was to complete a logical understanding regarding data warehousing enterprise. The conclusion from the survey was that data warehouse technology heavily raises financial and business returns in the adopters. They found the project team ability, Technical infrastructure, Technical architecture, Good vendor capability, clear objectives

issues for successful data warehousing enterprise.[211]

Joshi and Curtis in the paper entitled “Issues in building a successful data warehouse” in the executive’s journal, 1999, explore some key factor that any corporate should think about before planning to acclimatize data warehouse technology. Their most factors were Data warehouse development, architecture, User Access issues and Data issues. Based on reviewing the related research papers, they developed important factors that the organization must consider to have a successful planning of a data warehouse project [69].

D. Theodoratos, M. Bouzeghoub in their paper entitled “Data Currency Quality Factors in Data Warehouse Design” at International Workshop on Design and Management of Data Warehouses (DMDW’99), Heidelberg, Germany, 14. - 15.6. 1999 It described a DW system architecture that supports the performance quality goal and satisfies the data consistency and completeness quality goals.

In this framework it has addressed the problem of checking whether a view selection satisfies the given constraints. It have presented an approach for solving this problem that uses an AND/OR dag representation for multiple queries and views. Our approach allows also determining the change propagation frequencies that minimizes the view maintenance time cost and computes the optimal change propagation and query evaluation plans. More importantly, it can help the DW designer in the improvement of the selected view set, and can cooperate with DW design algorithms to generate a view set that satisfies the quality goals [185].

The accuracy and relevance of the data is essential to maintaining data warehouse quality. The timing of the import and frequency has a large impact on the quality as well [Ang & Teo, Management issues in data warehousing: insights from the Housing and Development Board (December 6, 1999)]. Data warehouse quality is easiest to maintain and support if the users are knowledgeable and have a solid understanding of the business processes [189].

Training the users to not only understand how to build queries, but also on the underlying data warehouse structure that enables them to identify inconsistencies much faster and to highlight potential issues early in the process [Ang & Teo, Management issues in data warehousing: insights from the Housing and Development Board (December 6, 1999)]. Any changes to the data tables, structure or linkages and the addition of new data fields must be reviewed with the entire team of users and support staff members in order to ensure a consistent understanding of the risks and challenges that might occur [189].

Data integrity is a concept common to data warehouse quality as it relates to the rules governing the relationships between the data, dates, definitions and business rules that shape the relevance of the data to the organization [Larry P., English, Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits (1999)]. [31,32]

Keeping the data consistent and reconcilable is the foundation of data integrity. Steps used to maintain data warehouse quality must include a cohesive data architecture plan, regular inspection of the data and the use of rules and processes to keep the data consistent whenever possible.

The easiest way to maintain data warehouse quality is to implement rules and checkpoints in the data import program itself. Data that does not follow the appropriate pattern not only be added to the data warehouse but require user intervention to correct, reconcile or change it frequently [Panos, Vassiliadis, Data Warehouse Modeling and Quality Issues (June 2000)]. In many organizations, these types of changes can be implemented only by the data warehouse architect, which greatly increases the data warehouse quality.[214]

The structure parameters adaptive neural tree algorithm for hierarchical classifications they use following operations for structure adoption: the making of new nodes, the divide of the nodes and the deletion of nodes. For input data, if the champion is a leaf and the error exceeds a predefined attention factor, a sibling node is formed. If the accumulated error of a leaf node goes over a threshold value over a period of time, the node will be dividing into several children. When a leaf node has not been used frequently over a period of time, the node will be deleted. Hebbian learning was used for the adaptation of parameters in structure parameters adaptive [257].

A self-organizing tree map algorithm, a new sub-node is put in to the tree structure when the distance between the input and engaging node is larger than the hierarchical control. Both these algorithms are designed to reduce the vector quantization mistake of the input data space, rather than for discovering the true hierarchical structure of the input data space. Data can be allocated not only to the leaf nodes but also to the internal nodes. Furthermore, there is no hierarchical relationship between upper level nodes and lower level nodes [258].

In structurally adaptive intelligent neural tree algorithm it creates a tree structured self organizing map. After higher level self organizing maps have been taught, a sub self organizing map is added to the node that has build up distortion error larger than a given threshold. These thresholds reduce with the growth of the tree level. The self organizing map tree is pruned by deleting nodes that have been inactive for a long time, and by merging nodes with high similarity [259].

A similar growing, hierarchical self-organizing is fixed lattice topology in each level of self organizing map for both of these latter algorithms makes them a smaller amount stretchy, and less striking.[260]

In dynamical growing self-organizing map algorithm is initialized with four nodes. A extend issue is used to control rising self-organizing map algorithm growth. If the whole error of a boundary node is larger than a growth threshold, a new boundary node is created and the growing self-organizing map algorithm enlarges. Hierarchical clustering can be achieved when several levels of growing self-organizing map algorithms with increasing division factor values are used. The final shape of the growing self-organizing map algorithm can represent the grouping in the data set [207].

According to HAC's algorithm which is based on data set and in this it is show the result lead to a hierarchical tree structure. The main thing in which, I observe that as the hight of tree expands, best technique is binary search. [208]
A growing cell structure algorithm is initial run on the data set. Then the divide of the growing cell structure result lead to a hierarchical tree structure. Similar to the HAC, the hierarchy of Tree growing cell structure is a binary tree [208].

According to this paper author give a new shining concept of self-organizing tree algorithm. But according to my opinion it is old concept but in a new words that is basic concept is binary tree(OLD) and new name is self-organizing tree algorithm [210].

Two classes of algorithms have been successfully used for the analysis of complex data (i.e., gene expression data). One is hierarchical clustering.

For example, Eisen et al. used hierarchical agglomerative clustering (HAC) to cluster two- spotted DNA microarrays data, and Wen et al. used HAC to cluster 112 rat central nervous system gene expression data;[157]

Dopazo et al. applied a self-organizing tree algorithm for clustering gene expression patterns. The alternative approach is to employ nonhierarchical clustering, such as Tamayo et al. used a self-organizing map to analyze expression patterns of 6,000 human genes [210].

In addition, Ben-Dor and Yakhini13 proposed a graph-theoretic algorithm for the extraction of high probability gene structures from gene expression data.[261]

while Smet et al. proposed a heuristic two-step adaptive quality based algorithm, and Yeung, K. Y. et al. have proposed a clustering algorithm based on probability models. All of these algorithms tried to find groups of genes that exhibit highly similar forms of gene expression [204,205].

Briggs shows negative aspect data ware house success. In short Those element which are responsible for mismanagement of data. [262]

1. first aspect which is mostly effect company attainment and capture, change in business environment, regulatory changes, organizational politics, and lack of senior management support. Others include human mistake, business traditions, organizational assessment making, change management, and lack of business drivers.

2. Second aspect include undervalue difficulty and workload, elevated expenditure, lack of understanding of outlay topic over time, difficulty in detail return on investment , and overselling benefits of data ware house. Others include the challenge of building a capable user developer management project panel, lack of IS staff political savvy, long development and delivery timeframe, poor selection of products and end-user tools, and failure to manage the scope of the project.

3. Third aspect include lack of general data definitions across different business units, data superiority, inefficient technology, data integration, and lack of clear understanding of DW applications and data needed.

The issue for the failure of data ware house. In view of issue of failure is important in set up undesired action for a achievement. According to expert, a DW project is a huge danger and is definitely endangered by several issues. These issues can be grouped into following categories, which are plan, technological, practical, and socio-technical. Each issue collection is described as follows: [214],

1. Plan factors stand by no standard or even broadly usual metadata

management method. Data production method, or plan methodologies for data ware house. Rather proprietary solutions from vendors or do it. It advice from experts seem to define the landscape.

2. Technological issues associate to the space that exists in the assessment and preference of hardware components.

3. Practical issues engage cause for lack concerning the use of the data ware house. Separately from traditional troubles in IS management, it is important to notice that the role of user society is vital. The end-user must be trained to the new technologies and included in the design of the warehouse.

4. Socio-technical issues meet on contravention the organizational agreement and are a outcome of the truth that the data ware house may reorganize the way the organization works and introduce the efficient or prejudiced domain of the stakeholders.

For example, imposing an exacting client device on users occupy the users' desktop, which is considered to be their personal region. Problems due to data rights and access are grouped into two categories.

First, data rights are power within an organization. Any attempt to share or take control over somebody else's data is equivalent with loss of power of this particular stakeholder.

Second, no division or department within an organization can claim to possess 100% clean, error-free data [18].

According to the study by Watson [138,139,140,141,142,143], the most common factors for the failure of a DW success include weak sponsorship and management support, insufficient funding, inadequate user involvement, and organizational politics.

There are several strategies to improve query response time in the data warehouse context: B+tree indexes are the most common supported structures in RDBMS, but it is a well-known fact that B+tree structures have inconveniences when the cardinality of the attribute is small. Another class of index structures, the bitmap indexes, try to overcome the problem by using a bit structure to indicate the rows containing specific values of the indexed attribute [240]. Although essential for the right tuning of the database engine, the performance of index structures depends on many different parameters such as the number of stored rows, the cardinality of the data space, block size of the system, bandwidth of disks and latency time, only to mention some [236].

Views pre-compute and store aggregates from the base data [Chaudury97]. The data is grouped using categories from the dimensions tables, which corresponds to the subjects of interest (dimensions) of the organization. Storing all possible aggregates poses storage space problems and increases maintenance cost, since all stored aggregates need to be refreshed as updates are being made to the source databases.

Many algorithms have been proposed for selecting a representative subset of the possible views [240], corresponding to the most usual query patterns. But the main problems associated with views are the difficulty to know in advance the expected set of queries, the problems of updating views to reflect changes made to base relations and the large amount of space required to store the views. It is worth noting that the techniques mentioned above (indexes and views) are general techniques that can (and should) be used in the data warehouse approach proposed in the present paper. In fact, in a distributed environment each individual machine must also be tuned and optimized for

performance.

A large body of work exists in applying parallel processing techniques to relational database systems with the purpose of accelerating query processing [229]. The basic idea behind parallel databases is to carry out evaluation steps in parallel whenever possible, in order to improve performance. The parallelism is used to improve performance through parallel implementation of various operations such as loading data, building indexes and evaluating queries.

One of the first works to propose a parallel physical design for the data warehouse was [228]. In their work they suggest a vertical partitioning of the star schema including algorithms but without quantifying potential gains. Partitioning a large data set across several disks is another way to exploit the I/O bandwidth of the disks by reading and writing them in a parallel fashion.

User queries have long been adopted for fragmenting a database in the relational, object-oriented and deductive database models [230,238,242]. That is, the set of user queries of a database is indicative of how often the database is accessed and of the portion of the database that is accessed to answer the queries. There are several ways to horizontally partition a relation. Typically, we can assign tuples to processors in a round-robin fashion (round-robin partitioning), we can use hashing (hash partitioning), or we can assign tuples to processors by ranges of values (range partitioning).Most of today's OLAP tools require data warehouses with a centralized structure where a single database contains all the data. However, the centralized data warehouse is very expensive because of great setup costs and lack of structural flexibility in data storage.

More importantly, "the world is distributed"': world-wide enterprises

operate in a global manner and do not fit in a centralized structure. Thus, a new paradigm is necessary. The first step in a new direction was the recent introduction of data marts, "small data warehouses" containing only data on specific subjects [235,236]. But this approach doesn't solve our problems of space and performance. Data marts provide more flexibility in the distribution of data but they still consist of static, self-contained units with fixed locations. By distributing small static portions of data to fixed locations, the system becomes more flexible, but on the other hand new problems arise, related to intra data mart communication, especially in what concerns the processing of queries. Many of today's data marts are basically standalone, because of the unsophisticated and rudimentary integration in the global data warehouse context.

In spite of the potential advantages of distributed data warehouses, especially when the organization has a clear distributed nature, these systems are always very complex and have a difficult global management [224]. On the other hand, the performance of many distributed queries is normally poor, mainly due to load balance problems. Furthermore, each individual data mart is primarily designed and tuned to answer the queries related to its own subject area and the response to global queries is dependent on global system tuning and network speed.

James Ang, Thompson S.H. Teo in paper entitled "Management issues in data warehousing: insights from the Housing and Development Board" at Elsevier, Decision Support Systems in 2000.In this paper, they examine data warehousing at the Housing and Development Board HDB[190], which is responsible for providing affordable, high-quality public housing to Singapore citizens. The HDB embarked on building a data warehouse because access to

the diverse and large amount of data in its operational systems, was becoming increasingly cumbersome and time consuming By building a data warehouse.

Panos Vassiliadis in their research "Data Warehouse Modeling and Quality Issues" at National Technical University of Athens in 2000 presented a set of results towards the effective modeling and management of data warehouse metadata with special treatment to data warehouse quality. The first major result was a general framework for the treatment of data warehouse metadata in a metadata repository. The framework requires the classification of metadata in at least two instantiation layers and three perspectives. The meta model layer constitutes the schema of the metadata repository and the metadata layer the actual meta-information for a particular data warehouse [215].

In the paper entitled "An empirical investigation of the factors affecting data warehouse success" in the MIS Quarterly in 2001, author by Wixom and Watson describe in their research, the following issues considered to be vital role play in the implementation of data warehouse these issues are Management support, Champion, Resources, User participation, Team skills, Source Systems, and Development technology. The results revealed that the following factors have a big and positive influence on the successful adoption of data warehouse project; a survey was used in this research to build up a model of data warehousing success [81].

Sudesh M. Duggal, Inna Pylyayeva in their research paper entitled "Data Warehouse – Strategic Advantage" at IACIS 2001 explain the Data warehouse functions as a conduit between the users and corporate data resources creating a flexible staging area for decision support applications. It provides a specialized decision support database that manages the flow of information from exiting

corporate database and external sources to end user to support strategic decision-making. To gain maximum value, an organization that relies on information should consider the development of a warehouse in conjunction with other decision support tools, such as Online Analytical Processing Systems, data mining, and Executive Information Systems. The warehouse should serve as the foundation for the vital process of delivering key information to analysts and decision-makers, which are the key users of the information [246].

Wixom et. all [25], presented an explanation of why some organizations realize more exceptional benefits than others after data warehouse installation. The authors started by giving a basic background about a data warehouse. Then they went through the obtainable benefits gained from data warehouse installation in general by the adopters. Three case studies of data warehousing initiatives, a large manufacturing company, an internal revenue service and a financial services company, were discussed within the context of the suggested framework.

The results from the case studies highlighted the benefits achieved by the three organizations. The researchers noticed that some of them considered more significant payoffs than the other adopters. The researchers built an argument about the main issues behind the success in the three cases. The argument led to the following issues: Business need, Champion, Top management support, user involvement, training matters, Technical issues (adequate tools), Accurate definition of the project's objectives, growth and upgradeability, Organizational politics, skilful team.

Mukherjee and D'Souza in their paper entitled "Think phased implementation for successful data warehousing, information systems

management", in 2003 presented a framework which might help the data ware house people to visualize how strategies can be included in each stage of data warehouse implementation process. They found that the data warehouse implementation process follows the three stage pattern of evolution (Pre-implementation, Implementation and Pos-Implementation phases). After reviewing previous related-research, a list of 13 critical implementation issues was developed; Data, Technology, Expertise, Executive sponsorship, Operating sponsorship, Having a business need, Clear link to business objectives, User involvement, User support, User expectation, organizational resistance, organizational politics, and Evolution and growth [89].

Hsin-Ginn Hwang, Cheng-Yuan Ku, David C. Yen, Chi-Chung Cheng in their research on " Critical factor influencing the adoption of data warehouse technology: A study of the banking industry in Taiwan", Decision support systems, 2004, Their focus scope was on the following packaged-dimensions (Organizational, Environmental, and Project dimensions). A questionnaire survey was designed and used to achieve the study's objective. A total of 50 questionnaires were mailed to CIOs in local banks [247].

After an intensive review of prior relevant studies, a total of ten factors influencing the success of data warehouse project were developed (Size of bank, Champion, Top management support, Internal needs, Degree of business competition, Selection of vendors, Skills of project team, organization resources, User participation, and Assistance of information consultants).After collecting the results from the questionnaire, they found that top management support, size of the bank, effect of champion, internal needs and competitive pressure would affect the adoption of data warehouse technology in banking industry in Taiwan.

Neil Warner in their paper entitle “Information Superiority through Data Warehousing” says Data warehousing has evolved through repeated attempts on the part of different researchers and organizations to give their organizations flexible, effective and efficient means of getting at the sets of data that have come to represent one of the organizations most critical and valuable property. Data warehousing is a tools that has grown out of the integration of a number of dissimilar tools and experiences over the last two decades. Data warehousing tools can provide an Enterprise Wide Framework for managing data within a Command and Control Organization [262].

Infrastructure is another important factor in data warehousing according to Gavin and Powell [249]. A data warehouse system can be functioning at the highest possible level for the available technology, but three or even only two years later, it can be considered obsolete. Improving the data warehouse architecture, both on a hardware level and a programming level, also can greatly increase data warehouse performance. Updating processors, adding additional storage space and using newer, more streamlined query protocols can greatly improve performance. In addition, these changes in overall data warehouse design can make a dramatic difference in the amount of data that can be stored as well as the speed at which the system can process individual queries.

Another approach that can help improve data warehouse performance is training. Data warehousing originally was designed to support decision making on a high executive level, but the overall usefulness of business intelligence has led to many other people using the data for a variety of purposes. In some cases, these employees have not received adequate training and do not know how to

construct efficient queries to retrieve the information they need. For these employees, training on the use of the system and how to effectively query the data can lead to great improvement in data warehouse performance. The main factors to consider when looking maintaining data quality: data integrity, data input source and methodology used, frequency of data import and audience.

The expert Solomon in their paper "Ensuring a successful data warehouse initiative" in 2005 it provide some guideline regarding the critical question that must be asked, some risk that should be weighted and some process that can be followed to help ensure a successful data warehouse implementation. it give two assumptions in first the guidelines here are for a large corporate data warehouse initiative that require either a new infrastructure environment or is significant enough to warrant analysis to determine whether it can comfortably integrated into an existing ,presumably extensible environment [249].

In second assumption it requirement in term of key performance indicator and specific analytical and business intelligence reporting need have been well established. The following are the guidelines that must be considered, by the organizations, to increase the chances for success Service level agreements and data refresh requirements ,Source system identification, Data quality planning, Data model design Extract, transform, and load tool selection must be there.

Hugh J. Watson and Thilini Ariyachandra in their research entitled "Data Warehouse Architectures: Factors in the Selection Decision and the Success of the Architectures" in July 2005 Based on this research, an overall architecture selection model is proposed. It takes the various selection factors and organizes

them into a causal-flow model. In this model, the need for information interdependence between organizational units and the nature of end user tasks combine to create the information requirements for the data warehouse. The information processing requirements and the source of sponsorship then combine to determine the view of the data warehouse [250].

The perceived ability of the IT staff, the availability of resources, and the urgency of need for the data warehouse combine as facilitating conditions for the selection of a particular architecture. And finally, the view of the warehouse and the facilitating conditions influence the architecture selection decision.

In paper entitled "An Investigation of the Factors Affecting Data Warehousing Success" authored by Roger L. Hayen, Cedric D.Rutashobya and Daniel E. Vetter in Issues in Information Systems Volume VIII, No. 2, 2007 state that DW has unique characteristics that may impact the importance of factors that apply to it. It was found that management support and adequate resources help address organizational issues that arise during DW implementations; resources, user participation, and highly-skilled project team members increase the likelihood that DW projects finish on-time, on-budget, and with the right functionality. The implementation's success with organizational and project issues, in turn, influences the system quality of the DW [251]

Jeff Theobald in "Strategies for Testing Data Warehouse Applications" publishes at Information Management Magazine, June 2007. This article provides practical recommendations for testing extract transform and load (ETL) applications based on years of experience testing data warehouses in the financial services and consumer retailing areas.

Every attempt has been made to keep this article tool-agnostic so as to be applicable to any organization attempting to build or improve on an existing data warehouse. In data warehousing, this is compounded because of the additional business costs of using incorrect data to make critical business decisions. To improve Data Warehouse Applications following strategies should be followed.

- Data completeness. Ensures that all expected data is loaded.
- Data transformation. Ensures that all data is transformed correctly according to business rules and/or design specifications.
- Data quality. Ensures that the ETL application correctly rejects, substitutes default values, corrects or ignores and reports invalid data.
- Performance and scalability. Ensures that data loads and queries perform within expected time frames and that the technical architecture is scalable.
- Integration testing. Ensures that the ETL process functions well with other upstream and downstream processes.
- User-acceptance testing. Ensures the solution meets users' current expectations and anticipates their future expectations.
- Regression testing. Ensures existing functionality remains intact each time a new release of code is completed.

Businesses are increasingly focusing on the collection and organization of data for strategic decision-making. The ability to review historical trends and monitor near real-time operational data has become a key competitive advantage [252].

Mark I Hwang; Hongjiang Xu in their paper “A Structural Model of Data Warehousing Success” in 2008 this paper contribute to the understanding

of data ware house success by showing the interrelationship among a set of variable. He state operational factor, technical factor, schedule factor, economic factor, system quality, information quality could vary as the time frame or the environment change [253].

A common organizational mistake in data warehousing is to ignore the currently owned technology and software capabilities and move ahead quickly, to purchase a different product [91] . An organization often has the toolset in place to effectively meet its data warehousing demands but simply needs the right partner to fully implement its current technology. The few simple rules that can help to develop a data warehouse on a small budget would include using what to have, using the knowledge, using what is free software and hardware buying only what have to, thinking and building in phases, and using each phase to finance or justify the remainder of the projects [Brian, Babineau, IBM Information Infrastructure Initiative Tames the Information Explosion (April, 2009)]. [61]

Nguyen Hoang Vu, Vivekanand Gopalkrishnan in research paper entitled "On Scheduling Data Loading and View Maintenance in Soft Real-time Data Warehouses" at 15th International Conference on Management of Data COMAD 2009, Mysore, India, December 9-12, 2009 expressed that an efficient technique aimed at updating data and performing view maintenance for real-time data warehouses while still enforcing these two timing requirements for the OLAP transactions. Through extensive empirical studies, they demonstrate the efficiency of ORAD in achieving the goal of building a real-time data warehouse [93]

In "Six tips for improving data warehouse performance" by Dr. Mark

Whitehorn, Co-Founder, Penguin Soft Consulting Ltd. in search data management express his view that any consultant worth his salt would look at the specific issues in a given company before suggesting changes to its data warehouse systems, here are top six tips for improving the performance of data warehouses:

1. Before even thinking about performance optimization, firstly identified the existing bottlenecks. If querying performance is CPU-bound, buying faster disks is a complete waste of money. Performance is often compromised when uses don't know the ins and outs of their particular database.
2. Use DELETE * to empty a table. That can be painfully slow compared with DROP and CREATE, which in turn is much slower than TRUNCATE. It might be worth hiring one from the standpoint of performance optimization alone.
3. In terms of querying, think about structuring the data as a MOLAP cube (i.e., a multidimensional online analytical processing one) and cranking up the aggregation until query performance flies. That may burn up disk space and data processing time, but it can make a huge difference.
4. The machine was "appropriated," the RDBMS was installed on the hard drive and a copy of the OLAP cube was created on the solid state drives (SSD). The cube aggregated much, much faster, but it was in querying that the most dramatic improvements were seen. Some of the queries ran 20 times faster with the same level of aggregation. The cost of the SSD was completely trivial (about $200) when compared with the improvement.
5. If possible, perform extract, transform and load (ETL) processing in memory. On a long ETL job, there may be virtue in caching to disk (in case the process fails), but try to keep the data in RAM during the

transformations. And cache to an SSD, not a hard drive.

6. Index analytical structures for analysis, not for transactions. The indexing strategy clearly was based on long experience with transactional systems and little familiarity with analytical ones [254].

RDBMS engines index the primary key of a table. That makes lots of sense in a transactional structure, where many of the queries use those indices – but very little sense in a star schema, where it is quite common for no, none, zero, nada queries to use the primary key indices. On the other hand, all of the analytical columns in dimension tables are highly likely to be searched and yet are often entirely bereft of indices.

Victor González Castro in their research thesis entitled “The Use of Alternative Data Models in Data Warehousing Environments” at Heriot-Watt University, Edinburgh, United Kingdom in may 2009 describe the Data Warehouses are increasing their data volume at an accelerated rate; high disk space consumption; slow query response time and complex database administration are common problems in these environments.

The lack of a proper data model and an adequate architecture specifically Help and support targeted towards these environments are the root causes of these problems. Inefficient management of stored data includes duplicate values at column level and poor management of data scarcity which derives from a low data density, and affects the final size of Data Warehouses. It has been demonstrated that the Relational Model and Relational technology are not the best techniques for managing duplicates and data scarcity. The novelty of this research is to compare some data models considering their data density and their data scarcity management to optimize Data Warehouse environments.

The Binary-Relational, the Associative/Triple Store and the Translational models have been investigated and based on the research results a novel Alternative Data Warehouse Reference architectural configuration has been defined. For the Translational model, no database implementation existed. Therefore it was necessary to develop an instantiation of its storage mechanism, and as far as could be determined this is the first public domain instantiation available of the storage mechanism for the Translational model [255].

Kimball, R in book entitled "The Data Warehousing Toolkit: Practical Techniques for Building Dimensional Data Warehouses" ISBN 0-471-15337-0, at Chapter first say that If the internal success factor is weak, then the likelihood is that there is an inefficient data model, not enough summarized data or the users are not well trained. If the external success factor is weak, then more resources should be put into addressing more of the needs. The success factor is therefore more than just an isolated statistic that can be reported to management, but is in itself a tool that can be used to make the data warehouse more effective for the enterprise [256].

According to Lauren, the absence or the lack of the following criteria is the common causes of data warehouse failures: pre- launch clear objectives or metrics, insider presence on data warehouse projects team, user demand for sophisticated data analysis, and initial involvement of business managers. In addition, other problems can arise: many major systems projects underway simultaneously, the CEO sets budget and deadlines before project team is on the board, and source data availability unconfirmed at the outset. Another potential Pitfall involves escalating user demand and unrealistic expectations [264].

According to Haisten one of the major causes of failures is that the database product was driving the project, not being driven by it. Others are the lack or the absence of close and effective link between business process analysis and data normalization, multi- national views available with the context, serious normalization, architecture for an integrated repository of brows able metadata, study of scale and capacity issues, and coordination with legacy application portfolio management [85].

According to Gupta, errors that data warehouse can contain involve data of four categories: incomplete, incorrect, incomprehensible, and inconsistent. Incomplete errors consist of missing records or missing fields. Incorrect data has wrong codes, calculations, aggregations, information entered into the system, pairing of codes, or duplicate records[82].

Incomprehensibility errors include multiple fields within one field, unknown code, many- to-many relationships that allow multiple patterns, and spreadsheets or word- processing files. Inconsistency errors include inconsistent use of different codes or meaning of a code, overlapping codes, different codes with the same meaning as well as inconsistent names, addresses, business rules, aggregating, timing, and use of an attribute, nulls, spaces, etc. If the data warehouse has that cannot be transformed properly, it is important for the data warehouse transformation process to use intelligent default values for the missing or corrupt data. It is also important to devise a mechanism for users of the data warehouse to be aware of these default values [9].

As data warehousing has become more and more important to businesses, increasing data warehouse performance has become vital. With many people depending on the data in the data warehouse to do their

jobs, data warehouse performance can have a profound impact on overall company performance [48,49].

Many companies rely on numerous ways to improve data warehouse performance, including clearing obsolete data, increasing storage space and improving overall data warehouse architecture and design, to keep the data warehouse and the company functioning at their best [265].

Data warehouse performance tends to degrade as more data is collected over a period of time. Increased data mining while important to the business increases the overall load on the system. More people making use of the system also increases the load as a larger number of queries are made by various employees. Removing obsolete information means that queries can be processed more quickly and return more relevant results, making overall data warehouse maintenance an important part of improving data warehouse performance.

According to Jorge Bernardino the distributed data warehousing affords several advantages to businesses, including suppliers and distributors. First and foremost, it is a cost-effective solution. By initially investing in a local Windows NT server, a department could build a small individual data store that could later be connected to additional ones, forming a "distributed" data warehouse. Instead of undertaking the painstaking task of integrating them all into one central store, the two or more could be virtually combined using network hardware and software [266].

A new approach for improving query response time in data warehouses named data warehouse striping. DWS can potentially achieve linear speedup and can significantly improve query response times. A simple but efficient algorithm for distributing the fact table solving the problems posed by its large volume. Takes advantage of the availability and low cost of microcomputers to interconnect these computers into networks which provide the distributed and parallel processing capabilities needed for handling very large data warehouses. This modular approach can be incorporated into a relational DBMS without significant modifications.

According to Comer Indexing techniques are among the first areas on which a database administrator will focus when good query performance in a read intensive environment is critical. Specialized indexing structures offer the optimizer alternatives access strategies for the time consuming full table scans. One of the most popular index structures is the B-tree and its derivatives [227].

Gerth Stølting Brodal in their research entitled “Cache Oblivious Search Trees via Binary Trees of Small Height” it is apparent that the effects of the memory hierarchy in today’s computers play a dominant role for the running time of tree search algorithms, already for sizes of trees well within main memory. It also appears that in the area of search trees, the nice theoretical properties of cache obliviousness seem to carry over into practice: in our experiments, the van layout was competitive with cache aware structures, was better than structures not optimized for memory access for all but the smallest n, and behaved robustly over several levels of the memory hierarchy[268].

One further observation is that the effects from the space saving and in-

crease in fan out caused by implicit layouts are notable. Finally, the method for dynamic cache oblivious search tree suggested in this paper seems practical, not only in terms of implementation effort but also in terms of running time.

Ortega-Binderberger studies the importance of user subjectivity and achieves query refinement through relevance feedback. Similarly, SODA presents several possible solutions to its users and allows them to like (or dislike) each result [269].

Elena Demidova [270] use query disambiguation techniques to process keyword queries automatically extracted from documents. he suggest a system that enables context-aware auto completion for SQL by taking into account previous query work-loads which are, in turn, represented as workload DAG.

When a user types a query, possible additions are suggested based on the highest ranked node in the DAG. Query suggestions include tables, views, functions in the FROM-clause, columns in the SELECT and GROUP BY clauses as well as predicates in the WHERE clause. The main difference to our approach is that Snip Suggest makes it easier for end-users to interactively build and improve SQL statements while SODA does not require any SQL knowledge at all. Moreover, SODA does also not rely on query logs.

Lukas Blunschi suggests that Search over Data Warehouse (SODA) is one step towards enabling end-users to interactively explore large data warehouses with complex schemas in a Google-like fashion. The key idea of SODA is to use a graph pattern matching algorithm to generate SQL based on simple key words.

In this experiments—with both synthetic data as well as with a large data warehouse of a global player in the financial services industry show that the generated queries have high precision and recall com-pared to the manually written gold standard queries. One of the strengths of SODA is that it can disambiguate the meaning of words by taking into account join and inheritance relationships among the matching tables. Moreover, SODA allows mitigating inconsistencies in the schema or data as well as data quality issues by updating the respective metadata graph or by extending the graph pattern match algorithm [271].

Azefack present an automatic, dynamic index selection method for data warehouses that is based on incremental frequent item set mining from a given query workload . The main advantage of this approach is that it helps update the set of selected indexes when workload evolves instead of recreating it from scratch. Preliminary experimental results illustrate the efficiency of this approach, both in terms of performance enhancement and overhead [220].

Eisen et al. applied a variant of the hierarchical average-linkage clustering algorithm to identify groups of co-regulated genome-wide expression patterns [157].

Loewenstein et al. applied agglomerative clustering to protein sequences that automatically builds a comprehensive evolutionary driven hierarchy of proteins from sequence alone [163].

Frey et al. applied an affinity propagation clustering technique to detect putative exons comprising genes from mouse chromosomes. While they claim lower computational cost in comparison to other algorithms, they do not include

the cost of pairwise similarity computations. Since this is the most expensive stage in large scale problems, the claimed advantage is exaggerated. Our focus is on reducing the computational complexity arising from this cost [158]. Frey et al. applied an affinity propagation clustering technique to detect putative exons comprising genes from mouse chromosomes. While they claim lower computational cost in comparison to other algorithms, they do not include the cost of pairwise similarity computations. Since this is the most expensive stage in large scale problems, the claimed advantage is exaggerated. Our focus is on reducing the computational complexity arising from this cost [158].

El-Sonbaty et al. [272] proposed an on-line hierarchical clustering algorithm based on the single-linkage method that finds at each step the nearest k patterns with the arrival of a new pattern and updates the nearest seen k patterns so far. Finally the patterns and the nearest k patterns are sorted to construct the hierarchical dendrogram. While they claim their method is sequential, at the arrival of each data item they compute similarity to all the data seen previously.Thus there is little computational saving in their method, and it is equivalent to re-training a new model at the arrival of new data. Contrast that with a truly sequential algorithm, where it is the model that is adapted, similar in fashion to the Kalman filter.

According to Bassam Farran, Amirthalingam Ramanan, and Mahesan Niranjan **i**n this paper they present an algorithm for on-line hierarchical clustering. The approach depends on the construction of an initial clustering stage using a random subset of the data. This establishes a scale structure of the input space. Subsequent data can be processed sequentially and the tree adapted constructively. They have shown that on small bioinformatics problems such an approach does not degrade the quality of the clusters obtained while saving

computational cost [273]

Their proposed technique could be significantly improved with an appropriate choice of the novelty threshold (θ). θ can be better estimated by taking into account the inter-cluster and/or intra-cluster information of the initial tree. This can be subsequently updated after the insertion of a newly arrived pattern. Another way of better estimating θ might be to use local thresholds associated with each parent or level of the tree, instead of a global threshold. The greatest benefit of the proposed technique lies in its application on very-large datasets.

Dr.N.Rajalingam [274] analyzes the performance of agglomerative and divisive algorithm for various data types. From this work it is found that the divisive algorithm works as twice as fast as that of agglomerative algorithm. It is also found that the time needed for string data type is high when compared to the other. The next observation is, in the case of binary field, the time needed to execute a two combined binary field is slightly larger or less equal to the time needed for single binary field.

It is also found that the running time get increased on an average of 6 times when the number of records get bled. More over the running time for all the agglomerative algorithms for same type of data and for same amount of records is more or less equal.

Chen et al. propose the incremental hierarchical clustering algorithm GRIN for numerical datasets, which is based on gravity theory in physics. In the first phase, GRIN uses GRACE, which is a gravity-based agglomerative

hierarchical clustering algorithm, to build a clustering dendrogram for the data sets [123].

Then GRIN restructures the clustering dendrogram before adding new data points by flattening and pruning its bottom levels to generate a tentative dendrogram. Each cluster in the tentative dendrogram is represented by the centroid, the radius, and the mass of the cluster. In the second phase, new data points are examined to determine whether they belong to leaf nodes of the tentative dendrogram.

Ester et al.[183,184] present a new incremental clustering algorithm called Incremental DBSCAN suitable for mining in a data warehousing environment. Incremental DBSCAN is based on the DBSCAN algorithm which is a density based clustering algorithm.

Due to its density-based qualities, in Incremental DBSCAN the effects of inserting and deleting objects are limited only to the neighborhood of these objects. Incremental DBSCAN requires only a distance function and is applicable to any data set from a metric space. However, the proposed method does not address the problem of changing point densities over time, which would require adapting the input parameters for Incremental DBSCAN over time.

Widyantoro et al. [195] present the agglomerative incremental hierarchical clustering (IHC) algorithm that also utilizes a restructuring process while preserving homogeneity of the clusters and monotonicity of the cluster hierarchy.

New points are added in a bottom-up fashion to the clustering hierarchy, which is maintained using a restructuring process performed only on the regions affected by the addition of new points. The restructuring process repairs a cluster whose homogeneity has been degraded by eliminating lower and higher dense regions.

Charikar et al. [6] introduce new deterministic and randomized incremental clustering algorithms while trying to minimize the maximum diameters of the clusters. The diameter of a cluster is its maximum distance among its points and is used in the restructuring process of the clusters.

When a new point arrives, it is either assigned to one of the current clusters or it initializes its own cluster while two existing clusters are combined into one. As indicated in the introduction, the clustering of a data stream is to some degree related to the incremental clustering of dynamically changing databases, although data streams impose different requirements on the mining process and the clustering algorithms.

The purpose of both methodologies is to provide the user with "up-to-date" clusters very quickly from dynamic data sets. However, when mining a dynamically changing database, the clustering algorithm has access to all points in the data base and not necessarily only the most recently inserted points, and the algorithm is not restricted to a sequential access to these new points. The clustering algorithm has slightly different requirements when mining data streams as explained next.

D. Barbara [275] outlines the main requirements for clustering data

streams. These requirements consist of 1) compact representation of the points that can be maintained in main memory even as lots of new points arrive, 2) fast incremental processing of new data points, and 3) clear and fast identification of outliers. The cluster assignment of new points should use a function that does not depend on comparison to past points and yet performs well.

Ganti et al. [276] also examine mining of data streams. A block evolution model is introduced where a data set is updated periodically through insertions and deletions. In this model, the data set consists of conceptually infinite sequence of data blocks D1, D2. that arrive at times 1, 2, ... where each block has a set of records. Some applications require mining all of the data encountered thus far (unrestricted window scenario), while others require mining only the most recent part (restricted window scenario) and updating the models accordingly.

The authors highlight two challenges in mining evolving blocks of data: change detection and data mining model maintenance. In change detection, the differences between two data blocks are determined. Next, a data mining model should be maintained under the insertions and deletions of blocks of the data according to a specified data span and block selection sequence.

The data stream model is also discussed by O'Callaghan et al. , who indicate that the model handles the following cases when mining dynamic data sets. First, a large portion of data arrives continuously and it is unnecessary or impractical to store all of the data. Second, the data points can be accessed only in the order of their arrival. Third, the data arrives in chunks that fit into main memory.

A new k-median algorithm called LocalSearch is presented to solve a k-median problem that minimizes the facility cost function, where the cost associated with each cluster is estimated by considering the sum of the square distance of the points to the centers of the clusters. The Stream algorithm is presented to cluster each chunk of the stream using the Local Search algorithm.

Aggarwal et al. [175]use the data summarization method BIRCH in the context of mining data streams. BIRCH compresses a dataset into so-called clustering features (CFs). A clustering feature CF = (n, LS, SS), where LS is the linear sum of the n points compressed by the CF and SS is their square sum. The scheme presented by

Aggarwal et al. [175] for clustering data streams combines online micro clustering with offline macro clustering . During the micro clustering phase, a temporal version of the clustering features of BIRCH and pyramidal time frames are used to store on disk micro clusters from different time snapshots in a pyramidal pattern. Once the user specifies the window for mining the macro clusters, the micro clusters for that window are extracted using the additively property of the clustering features and the macro clusters are uncovered using a modified k-means algorithm that regards the micro clusters as points.

Zobel and Moffat [87] provided a set of standard similarity measures that include combining and weighting functions. In this research, M-trees are not used for indexing spatial data objects, but they are used for clustering data objects which are then reconcile. Like B-trees and R-trees, M-trees grow in a bottom-up fashion and are balance. The M-trees differ from B-trees in the fact

that a node in M-trees is split when it is 100% full whereas a node in B-trees is split when its capacity reaches a certain threshold (usually when it is approximately 70%full).

Tasawar et al.,[277] proposed a hierarchical cluster based preprocessing methodology for Web Usage Mining. In Web Usage Mining (WUM), web session clustering plays a important function to categorize web users according to the user click history and similarity measure.

Web session clustering according to Swarm assists in several manner for the purpose of managing the web resources efficiently like web personalization, schema modification, website alteration and web server performance. The author presents a framework for web session clustering in the preprocessing level of web usage mining. The framework will envelop the data preprocessing phase to practice the web log data and change the categorical web log data into numerical data. A session vector is determined, so that suitable similarity and swarm optimization could be used to cluster the web log data. The hierarchical cluster based technique will improve the conventional web session method for more structured information about the user sessions.

Yaxiu et al.,[278] put forth web usage mining based on fuzzy clustering. The World Wide Web has turn out to be the default knowledge resource for several fields of endeavor, organizations require to recognize their customers' behavior, preferences, and future requirements, but when users browsing the Web site, several factors influence their interesting, and various factor has several degree of influence, the more factors consider, the more precisely can mirror the user's interest. This paper provides the effort to cluster similar Web user, by involving two factors that the page-click number and Web browsing time that are stored in the Web log, and the various degree of influence of the

two factors. The method suggested in this paper can help Web site organizations to recommend Web pages, enhance Web structure, so that can draw more customers, and improves customers' loyalty.

Web usage mining based on fuzzy clustering in identifying target group is suggested by Jianxi et al.,[279] Data mining deals with the methods of non-trivial extraction of hidden, previously unknown, and potentially helpful data from very huge quantity of data. Web mining can be defined as the use of data mining methods to Web data. Web usage mining (WUM) is an significant kind in Web mining. Web usage mining is an essential and fast developing field of Web mining where many research has been performed previously. The author enhanced the fuzzy clustering technique to identify groups that share common interests and behaviors by examining the data collected in Web servers.

Houqun et al.,[280] proposed an approach of multi-path segmentation clustering based on web usage mining. According to the web log of a university, this paper deals with examining and researching methods of web log mining; bringing forward a multi-path segmentation cluster technique, that segments and clusters based on the user access path to enhance efficiency.

B+tree indexes are the most common supported structures in RDBMS, but it is a well-known fact that tree structures have inconveniences when the cardinality of the attribute is small. Another class of index structures, the bitmap indexes, try to overcome the problem by using a bit structure to indicate the rows containing specific values of the indexed attribute by O'Neil in 1997 [241].

According to Jurgens the performance of index structures depends

on many different parameters such as the number of stored rows, the cardinality of the data space, block size of the system, bandwidth of disks and latency time, only to mention some. The use of materialized views (or summary tables) is probably the most effective way to speedup specific queries in a data warehouse environment. Materialized views pre-compute and store (materialize) aggregates from the base data [237].

According to Chaudury in 1997 The data is grouped using categories from the dimensions tables, which corresponds to the subjects of interest (dimensions) of the organization. Storing all possible aggregates poses storage space problems and increases maintenance cost, since all stored aggregates need to be refreshed as updates are being made to the source databases. Many algorithms have been proposed for selecting a representative subset of the possible views for materialization [267].

Ezeife et all said that corresponding to the most usual query patterns. But the main problems associated with materialized views are the difficulty to know in advance the expected set of queries, the problems of updating materialized views to reflect changes made to base relations and the large amount of space required to store the materialized views. It is worth noting that the techniques mentioned above (indexes and materialized views) are general techniques that can (and should) be used in the data warehouse approach proposed in the present paper. In fact, in a distributed environment each individual machine must also be tuned and optimized for performance [230].

According to DeWitt a large body of work exists in applying parallel processing techniques to relational database systems with the

purpose of accelerating query processing. The basic idea behind parallel databases is to carry out evaluation steps in parallel whenever possible, in order to improve performance. The parallelism is used to improve performance through parallel implementation of various operations such as loading data, building indexes and evaluating queries. One of the first works to propose a parallel physical design for the data warehouse [265].

Datta in 1998 in their work he suggest a vertical partitioning of the star schema including algorithms but without quantifying potential gains. Partitioning a large data set across several disks is another way to exploit the I/O bandwidth of the disks by reading and writing them in a parallel fashion. User queries have long been adopted for fragmenting a database in the relational, object-oriented and deductive database models [228].

According to Ezeife the set of user queries of a database is indicative of how often the database is accessed and of the portion of the database that is accessed to answer the queries. There are several ways to horizontally partition a relation. Typically, it can assign records to processors in a round-robin fashion (round-robin partitioning), It can use hashing (hash partitioning), or it can assign records to processors by ranges of values (range partitioning). Most of today's OLAP tools require data warehouses with a centralized structure where a single database contains all the data. However, the centralized data warehouse is very expensive because of great setup costs and lack of structural flexibility in data storage [231].

More importantly, "the world is distributed"': world-wide enterprises operate in a global manner and do not fit in a centralized structure. Thus, a new paradigm is necessary. The first step in a new direction was the

recent introduction of data marts, “small data warehouses” containing only data on specific subjects But this approach doesn’t solve our problems of space and performance.

Data marts provide more flexibility in the distribution of data but they still consist of static, self-contained units with fixed locations. By distributing small static portions of data to fixed locations, the system becomes more flexible, but on the other hand new problems arise, related to intra data mart communication, especially in what concerns the processing of queries.

Many of today’s data marts are basically stand-alone, because of the unsophisticated and rudimentary integration in the global data warehouse context. In spite of the potential advantages of distributed data warehouses, especially when the organization has a clear distributed nature, these systems are always very complex and have a difficult global management.

Albrecht propose [224] the performance of many distributed queries is normally poor, mainly due to load balance problems. Furthermore, each individual data mart is primarily designed and tuned to answer the queries related to its own subject area and the response to global queries is dependent on global system tuning and network speed.

2.2 Performance Management

It is important to remember the success of a Data Warehouse can only be dignified once the data is loaded and users are able to make corporate level decisions by extracting data from the Data Warehouse. This related information describes the atmosphere on any Data Warehouse project at any stage. There are so many unknowns throughout the process be it loading, query work load and future work the database administrators life is one of continual uncertainty. This process of continual uncertainty is not necessarily a bad thing either. The reason for this can be determined from the need for a Data Warehouse. If the users knew everything about the contents of their data they would not need a Data Warehouse.

The performance of a Data Warehouse is largely a function of the quantity and type of data stored within a database and the query/data loading work load placed upon the system. When designing and managing such a database there are numerous decisions that need to be made that can have a huge impact on the final performance of the system.

This makes the Data Warehouse performance management process extremely difficult as the workloads very often cannot be predicted until finally the system is built for the first time and the data is in a production status.

As a system goes production for the first time only then may a system administrator discover there are performance problems. If there are too many performance problems in the running of the Data Warehouse the viability of the project becomes marginal or questionable.

The workload on a data warehouse hardly ever remains fixed. New users carry different kinds of demands, existing users change their focus and often the depth of their studies, the business cycle presents its own kinds of peaks and valleys, and in most cases the data warehouse expands as it stores data to cover longer periods of time. Given the dynamic nature of modern distributed environments, both source data updates and schema changes are likely to occur autonomously and even concurrently in different data sources.

Sometimes availability is overlooked in the evaluation of performance. If a system is unavailable, then it is not performing. Therefore, the ability of the platform to deliver the required availability is critical. Some might question the need for high availability of a warehouse compared to the availability requirements of an operational system. A warehouse may, in fact, need 24x7 availability. Data warehouses are typically updated only periodically, in a batch fashion, and during this process the warehouse is unavailable for querying. This means a batch update process can reorganize data and indexes to a new optimal clustered form, in a manner that would not work if the indexes were in use.

In this simplified situation, it is possible to use specialized indexes and materialized aggregate views (called summary tables in data warehousing literature), to speed up query evaluation. Consider that queries against a warehouse have to process large amount of data, which may take longer time. Longer outages may be accept by an prepared system if they are planned around user queries, but unplanned outages at the middle or end of a long running query may be unacceptable for users.

In addition, the more a company uses a warehouse to make strategic business decisions, the more the warehouse becomes just as critical as the operational systems that process the current orders. Many have argued the value

of a warehouse to project future buying patterns and needs to ensure that the business remains competitive in a changing market place. These decisions affect whether there be future orders to record in an operational system.

2.3. Performance tuning strategies

The art of performance tuning has always been about matching workloads for execution with resources for that execution. Therefore, the beginning of a performance tuning strategy for a data warehouse must include the characterization of the data warehouse workloads. To perform that characterization, there must be some common metrics to differentiate one workload from another.

One of the problems with data warehousing is that organizations are rushing to build it without regard to how that impact existing systems architectures, or how it be integrated with other applications. Critics also contend that warehouses ignore process and function, essentially sterilizing the data and removing the application context by isolating it in one or more relational database engine.

Performance issues in data warehousing are centralized around access performance for running queries and incremental loading of snapshot changes from the source systems. The following seven concepts can be considered for a better performance:

1. Query Performance
2. Coordination and Communication
3. Education, Training and documentation
4. Help and support
5. Management of data flow on network

6. Loading and Cleansing Data
7. Software and Hardware
8. Materialized view

2.3.1 Query Performance

Modern database systems can greatly benefit from query performance. the execution latency of a query plan on a given hardware and system configuration. For ex-ample, resource managers can utilize Query performance to perform workload allocation such that interactive behavior is achieved or specific Query targets are met. Optimizers can choose among alternative plans based-on expected execution latency instead of total work incurred.

Accurate Query performance is important but also challenging: database systems are becoming increasingly complex, with several database and operating system components interacting in sophisticated and often unexpected ways. The heterogeneity of the underlying hardware platforms adds to this complexity by making it more difficult to quantify the CPU and I/O costs.

Analytical cost models predominantly used by the current generation of query optimizers cannot capture these interactions and complexity; in fact, they are not Data warehouses usually contain a huge amount of data that must be analyzed and, provide that analysis, helping in the organizational decision making process. The success of this kind of support depends greatly on database systems and correlated analysis tools.

Data warehouses differ significantly from the traditional database applications. Data warehouses provide a different context in which huge amounts of data must be processed efficiently and queries are often complex,

but still require interactive response times. In data warehouse environments the data is used for decision support and large sets of data are read and analyzed. many tools and techniques have increased in the performance of database management as data warehousing and data mining. These warehouses provide storage functionality and responsiveness to queries beyond capacity of transaction oriented database.

One of the most important requirements of a data warehouse server is the query performance. The principal aspect from the user perspective is how quickly the server processes a given query: “the data warehouse must be fast”. The main focus of our research is finding adequate solutions to improve query response time of typical data ware house queries and improve scalability using an environment that takes advantage of characteristics specific to the data ware house context. Our propose model provides very good performance and scalability even on huge data warehouses.

There are many solutions to speed up query processing such as summary tables, indexes, parallel machines, etc. The performance when using summary tables for predetermined queries is good. However when an unpredicted query arises, the system must scan, fetch, and sort the actual data, resulting in performance degradation. Whenever the base table changes, the summary tables have to be recomputed. Also building summary tables often supports only known frequent queries, and requires more time and more space than the original data. Because we cannot build all possible summary tables, choosing which ones to be built is a difficult job.

Moreover, summarized data hide valuable information. Indexes are database objects associated with database tables and created to speed up

access to data within the tables. Indexing techniques have already been in existence for decades for the OLTP relational database system but they cannot handle large volume of data and complex and iterative queries that are common in OLAP applications.

Once data has been cleaned properly they are to be stored in large storages such as functional data bases i.e. Data ware houses. Only to dump data in storage will again create jumbled type of data. In both cases faster search techniques need to be evolved for better query processing. To arrange data in a database in such a way that retrieval/accessing and updating becomes easier and faster, a process known as indexing comes in. An index access structure is similar to the index used in a text book which lists important terms at the end of the book in alphabetical order along with a list of address page numbers.

With the emergence of powerful new indexing technologies, instant and ad-hoc queries and fast data analysis are possible using existing databases. Despite the good customer service and data analysis capabilities, many customer services and data warehousing applications lack good performance. To meet this challenge in business applications such as customer services, e-commerce etc., data warehouses must deliver data quickly through user friendly methods.

The main purpose of data is to access and use it. Quick access of information requires storage of data in structured form. The storage of data in structured form helps develop efficient and faster search technique to handle more complex queries and retrieve data with maximum precision. The evolution of storage and access technique starts with the evolution of flat files. This was

suitable, when files are small. The flat files require sequential scan of all the records in the file. the data access/retrieval time increases with the increase in the volume of data and thus results in more costly processing.

Accurate query performance is central to effective resource management, query optimization and user experience management. Analytical cost models, which are commonly used by optimizers to compare candidate plan costs, are poor analyst of execution latency.

The center of research on query performance calculation is to calculate the electiveness of a query given a search system and a collection of documents. if the performance of queries can be predictable in advance of, or during the retrieval stage, particular measures can be taken to improve the overall performance of the system. In particular, pre retrieval predictors predict the query performance before the retrieval step and are thus in dependent of the ranked list of results; such predictors base their predictions solely on query terms.

2.3.2 **Coordination and Communication**

A communication process also offers the data warehouse team to measure progress and identify and resolve issues before they become a serious problem. The communications program provides the business components with increased capabilities and functions. Communication is the medium or the process by which one can convey or express his thoughts, views and feelings. Whatever be the mode of communication, the effectiveness of the communication is very important for the success of an individual or a team.

Coordination and communications program keeps people informed and

generates interest in using the data warehouse. Organizational rules to rationalize inconsistencies had to be established. Clear and consistent communication of company-wide warehouse goals and policies fosters employee participation on three critical issues: First, it reinforces the front-line employees' contribution of information to the warehouse. Second, it encourages information sharing to support ongoing business activities. Third, it inspires middle management to use the data warehouse to inform key stakeholders regarding decisions, and new projects.

Once the communications framework was established the information was then placed in the hands of skilled logisticians. Using a report generator, the information is provided to decision makers with the expertise to analyze the results and recommend appropriate action.

The scope of the selected communication program identifies the people who should be contacted, the main messages to pass, and the type of communication and its frequency [30].This can be achieved by giving a detailed and scheduled program of education and training for developing and supporting vision clarity for the data warehousing environment.

Educating the knowledge workers about the purpose and benefits of a real and complete data warehouse and feedback and suggestions from data warehouse users can also help in improving data warehouse performance. By providing a place or group to contact outside the helpdesk to address concerns or to act as a contact point for marketing data warehousing services to new business units and involving other interested parties in data warehouse planning, analysis and design sessions, users understanding of data warehouses can be greatly increased.

Hence the key factor for success is communication, especially effective

communication.

But the effective communication important because with the help of this we can improve data ware house performance.

1. ***Communication help strength relationships and improve bonding among the people in a team or organization***.

The more communicate openly and frankly, the listener tends to build his levels of comfort and confidence. Once the confidence level increases among the team members, the goal turns into a common one and thus can be easily achieved.

2. ***Effective communication prevents many misunderstandings in general***.

Assumption is a major adversity as many people tend to think that others know what they are thinking. But when opinions clash, it turns into misunderstandings and conflicts. Such problems can be resolved by taking time to convey across thoughts and feelings in an accurate manner. Thus communicating is important so as to make sure that everyone is in synchronization to avoid misunderstanding and conflicts.

3. ***It is a must that important decisions and factors are documented for reference in future***.

This kind of written communication can be held as a proof for future use. Also such kind of documentation helps in keeping the team members in synchronization with the goals and standards that were formulated even before they joined the team.

4. ***Communication is always not necessarily from the top management***.

It also needs to be equipped by every member of the team to send their thoughts and views. When there are cases of complaints or misunderstandings, team members need to communicate that to their leaders following the right hierarchy so as to reduce the level of frustration and friction within the team.

5. ***Decision making in an organization hugely depends on the thoughts and ideas that are communicated from all the employees***.

Strategies like brainstorming and other kind of decision making tools need thoughts and views of all the involved members as the prime input. Such kind of methods requires effective communication processes to make the entire process a success.

6. ***In a huge team, it is mandatory that all the team members are on the same levels in terms of information knowledge and messages conveyed***.

In these scenarios, team meetings and discussions help resolve misunderstandings and communication gaps. So to ensure the success in a team effort, the information needs to be effectively communicated to all the members. Oral communication followed by verbal communication ensures that the information passed to all the members is same and accurate.

7. ***Job delegation and responsibility assigning is a frequently encountered act in organizations and teams***.

But if the job details are not effectively passed on, the work not be completed satisfactorily and would cause failure of the job. Thus communication plays an important role again in this regard. Communicating the job details accurately is important and necessary so as to get the job done as per requirements.

Thus based on the above factors and discussions, we can conclude that communication is not just a necessary ingredient for success but is also needed for maintaining harmony and a peaceful professional relationship within the data ware house organization.

2.3.2. **Education, Training and Documentation**

The quality of employees and their development through training and education are major factors in determining long-term profitability of a small business. If hire and keep good employees, it is good policy to invest in the development of their skills, so they can increase their productivity. Training often is considered for new employees only. This is a mistake because ongoing training for current employees helps them adjust to rapidly changing job requirements.

Training expands the communication process by maintaining a level competence in both the business and IT community as to the tools and mechanisms of the data warehouses.

Training needs to be focused on data warehouse concepts and terminologies, introduction to the organizational data, where is that located in the warehouse and how it is related to the reports or systems user already is using, the mechanics of using the tool. It is important for people to understand basic navigation within the tool.

The type of analysis that can be performed and how use the tool against the data. What starter set of reports has been developed, how to use them and how they are organized. Within data warehousing there are a number of major topics and an even larger number of secondary topics. In addition

there are a series of related subject matters that can be covered from the perspective of data warehousing such as object-oriented technologies client/server technologies and the internet.

An important component of the roll-out plan is the training schedule. To create a training schedule, consider the following:

- Goals and Objectives of the training,
- Number and types of users,
- Number of trainers,
- Number of training facilities,
- Number of training modules,
- Time required for each training module,
- Vendor-supplied training for purchased tools and products.

Training objectives are one of the mainly essential parts of training program. While a number of people believe of training objective as a misuse of precious time. The counter argument here is that assets are always limited and the training objectives actually lead the design of training. It provides the clear strategy and develops the training program in less time because objectives center specifically Help and support on needs. It helps in adhering to a plan.

Training objective tell the trainee that what is projected out of him at the end of the training program. Training objectives are of great significance from a number of stakeholder perspectives,

1. Trainer
2. Trainee
3. Designer

4. Evaluator

Trainer – The training objective is also beneficial to trainer because it helps the trainer to measure the progress of trainees and make the required adjustments. Also, trainer comes in a position to establish a relationship between objectives and particular segments of training.

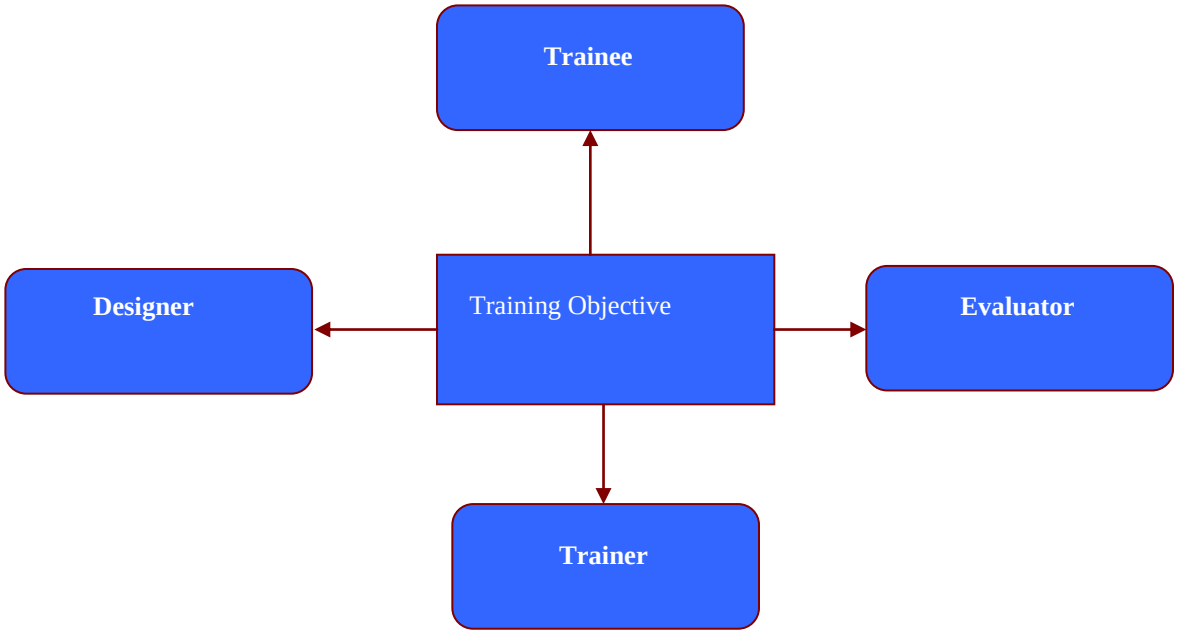

Fig. 2.1 Role of Trainer, Trainee, Designer and Evaluator

Trainee – The training objective is beneficial to the trainee because it helps in reducing the anxiety of the trainee up to some extent. Not knowing anything or going to a place which is unknown creates anxiety that can negatively affect learning. Therefore, it is important to keep the participants aware of the happenings, rather than keeping it surprise.

Secondly, it helps in increase in concentration, which is the crucial factor to make the training successful. The objectives create an image of the training program in trainee's mind that actually helps in gaining attention.

Thirdly, if the goal is set to be challenging and motivating, then the likelihood of achieving those goals is much higher than the situation in which no goal is set. Therefore, training objectives helps in increasing the probability that the participants be successful in training.

Designer

The training objective is beneficial to the training designer because if the designer is aware what is to be achieved in the end then he'll buy the training package according to that only. The training designer would then look for the training methods, training equipments, and training content accordingly to achieve those objectives. Furthermore, planning always helps in dealing effectively in an unexpected situation. Consider an example; the objective of one training program is to deal effectively with customers to increase the sales. Since the objective is known, the designer design a training program that includes ways to improve the interpersonal skills, such as verbal and non verbal language, dealing in unexpected situation i.e. when there is a defect in a product or when a customer is angry.

Therefore, without any guidance, the training may not be designed appropriately.

Evaluator

It becomes easy for the training evaluator to measure the progress of the trainees because the objectives define the expected performance of trainees. Training objective is an important to tool to judge the performance of

participants.

It should be noted that in each category there may be some individuals who already have some level of data warehousing knowledge and related concepts and thus may start at a different point in the curriculum.

The optimal learning environment may be a customized class created either internally or by an end or, using a subset of the company's own data [39]. The advantages this approach has are listed below:

1. Uses data that users know and can identify with (company's own data).
2. Provides custom training material and manuals.
3. Provides formal exercises and materials to assist in training new personnel.

It should be noted that one day long training at the vendor site is not enough for an average user of the data warehouse. The tools used to extract information from a data warehouse are extremely sophisticated. Often users get confused by the overload of information or forget the information before having a chance to use it. It is imperative that procedures and programs be implemented that can provide continuous help and assistance on the data warehouse and the front end tool.

Training is an essential component of most industrial pursuits [26]. It is the process whereby an individual acquires the necessary skills to perform a specific job or carry out specific tasks . In his landmark book on data warehousing, the data warehouse toolkit; Ralph Kimball insists to follow the following points for an effective education program [28]:

1. Understand the target audience. Don't overwhelm.

2. Don’t train the business community early prior to the availability of data and analytic applications.
3. Postpone the education if the data warehouse is not ready to be released.
4. Gain the sponsor’s commitment to ‘no education, no access’ policy.

Without proper training the intended users would not be able to take full advantage of the capabilities of the data warehouse [30]. A training program should be started for medium or large scale data warehousing projects. These types of projects are delivered in multiple iterations therefore the program is essential for the success of the data warehousing project. Data mart projects that overlap the business or where the focus is on corporate or executive information should also consider implementing a training program as a required infrastructure process.

To establish an efficient and result oriented training program one needs to establish a training team and develop a financial plan for budgeting [30]. The members of the team should establish a list of recommended technology components and standards that should be taught. The team further reviews and approves the proposed approach, functional architecture and technology components and standards.

The next responsibility of the team is to design the data warehouse training program structure. The team analyzes current training channels and mechanisms, defines vendor to internal IT involvement and responsibility in this process, determines which training components to buy and which to develop in-house, conduct the vendor selection evaluation process, defines the timing and frequency of the proposed training process, determines the required supporting organizational structure and conduct training developer

team training, designs and tests the required organizational structure, roles and responsibilities, deliverables, and procedures for the training program and required interfaces to the data warehouse project development life cycle, Help and support Center function and program office. After going through all these tasks the team finally compiles a proposed implementation and support plan.

The superiority of workers and their growth through training and education are major issues in determining enduring prosperity of an undersized business. If hire and keep good employees, it is good rule to invest in the development of their talent, so they can increase their productivity.

Training often is considered for new employees only. This is a mistake because ongoing training for current employees helps them adjust to rapidly changing job requirements.

Benefits of training are that a small business receives from training and developing its workers, including:

- Increased productivity.
- Reduced employee turnover.
- Increased efficiency resulting in financial gains.
- Decreased need for supervision.

Employees frequently develop a greater sense of self-worth, dignity and well-being as they become more valuable to the firm and to society. Generally they receive a greater share of the material gains that result from their increased productivity. These factors give them a sense of satisfaction through the achievement of personal and company goals. In any organization staff can be dividing in three levels.

Level 1 - The employees with little or no knowledge of computers.

Level 2 - Employees can operate a computer system, have basic knowledge of databases and networking and can make reports using the computer systems.

Level 3 - These are professional computer specialists and are liable for the management, maintenance and development of operational systems used by organization.

It is the responsibility of organization to provide education to level 1 and 2 employees, as their skill in training program. So they can work with smoothly.

The next step is the implementation of the data warehouse training program. This includes confirming completeness of all designed components, processes, human resources and facilities and preparing implementation of the training program in support of data warehousing efforts, selecting and training delivery of user team members, setting up facilities, reporting, and feedback procedures and the supporting desktop environment for the training program. The training program is reviewed from time to time for betterment.

As a result of the efforts of the training team members, a lot of documents, procedures, facilities, and systems capabilities are developed to deliver the data warehouse training program. These include a training program charter or mandate containing program goals and objectives, budget, organizational structure for training team, standards to be followed and the program procedures to deliver training. Additionally documents pertaining to training program facility locations and staffing are also produced.

2.3.3 Help and Support

The Help and support Center is an important division for any organization as it serves as the primary interaction point between customers and the company. In many situations, it is the only interaction point and therefore, responsible for the customer's experience and satisfaction. Due to this heightened level of importance, it is critical that the contact handling process is conducted both efficiently and effectively.

In order to create useful customer interaction, it has to work personally with customers to understand and react to their needs. Development customers translates into achievement, and using a Help and support Center helps achieve this goal. A Help and support Center can be a qualified extension of business, and it creates a reliable qualified presence of company from a customer's perspective. Help and support Centers can help company interact more with customers. If customers are consistently getting an on time response from company, they are likely to feel prized and continue to do business with them.

Help and support Counters have a well skilled staff, and can communicate successfully with customers about goods, services, and business. A Help and support Counters can help sustain and make customer profiles, so that can constantly learn about them. A Help and support Counters staff and systems can gather and examine information about customers, and with this knowledge, this can make decisions based on relationship with them. Help and support Counters have the technology to sector the customers and then provide user suitable interfaces, routing routines, service levels and content. This provides the opportunity for maximum efficiency with customer transactions.

Building a business is not always about new customers. Help and

support Centers can also analyze customer profiles and anticipate customer requests. This can increase cross-sell and up-sell opportunities by offering suggestions to customers who are more likely to make additional purchases from company. By using a Help and support Center, company is perceived as easy to do business with because of the responsiveness to customers. This increased value to the customers means the likelihood of their retention.

A Help and support Center can help to focus on these customer relationships by increasing time on customer Help and supports. In order to ensure success one needs to develop a support structure and plan an approach. When people are using the system, the questions flow. If there are no questions than it is likely that no one is using the system. The question asked could be about the validity of the data itself, how to use the data, what calculations make sense and what levels of aggregation are valid, how to pick a report, how report measures were calculated, how to use the applications, how to change an application, and how to build and own application etc.

User support is crucial immediately following the implementation in order to ensure that the business community gets hooked . For the first several weeks following user education, the support team should be working proactively with the users. It can't sit back and assume that no news from the community is good news. If there is nothing heard from the business users it means that no one is using the data warehouse. In such a case the support professionals should turn up to the business community so that the users of the data warehouse have easy access to support resources. If problems with the data or applications are uncovered, immediately try to rectify the problems. Again if the warehouse deliverables are not of high quality, the unanticipated support demands for data reconciliation and application rework can be devastating.

A Help and support Center acts as an extension to the user and as an assistant to the data warehouse support group. For the end user the Help and support Center addresses a number of issues that include:

1. Security and sign on (from the client network, or remote)
2. Access to detail or aggregated data stored in a warehouse, operational data store or data mart
3. Data staging or data quality management problems
4. Ad hoc query processing
5. Predefined and scheduled reports or automated routine processing
6. System response time

For the data warehouse IT support and monitoring team, the Help and support Center quickly points out challenges and issues including installation of new functionality, certification of new hardware or software, facility or network capacity thresholds, and scheduled job performance and environment backups etc.

The Help and support Center based on its authority and expertise of its staff members, resolves, and redirects or escalates a problem event that has occurred. Help and support Center services are provided in a number of ways including phone, intranet or internet based help, in person or through the automated help facilities available in the data warehouse.

The first and most obvious method of support is to create and expand the Help and support Center. This gives the user's one place to Help and support when help is needed. The people at the Help and support Center needed to be able enough to solve a technical problem themselves, but also need to have an understanding of the business, the data that is in the warehouse and how it can/should be used. This complete skill set may not reside with one single

person, instead a support team may be needed that can control the situation.

Another role of the data warehouse support and protection group is the problem resolution when some problem is encountered in the data warehouse. A Help and support Center acts a coordinating body for not only collecting and logging problems with the data warehouse environment but also determining where future requirements may lay.

In some organizations the data warehouse help service is seen as an extension to an existing OLTP Help and support Center based program while in others a separate organization is struck to deal with the distinctive nature of this environment. Within this context, the problem management process is the vehicle for recording and resolving issues as they arise, again pointing the way towards future improvements or the need for new functionality in the data warehouse. Problems can be protection, enhancement based, or they can point the way towards new development.

The problem management process much like the project change management process, acts as a vehicle for initiating and determining the nature of work for our protection, enhancement, or new development data warehousing projects.

Help and support Center services are provided from a central Help and support Center or they are distributed to the various business locations. If more hands on, or personal support is required especially during the early months of a major data warehouse project, this support is usually phased out of the data warehouse group or centralized over time.

Other options include ‘train the trainer’ approach where local representatives are given more extensive product and application based training than the average end user. Other options for providing Help and support Center services include identifying local ‘power users’, those more in tune with the technology or using a much broader band of the available services.

These people can be tapped in a backup or support role to the Help counter. To keep them interested, these talented staff can be offered ‘first look’ and more proactive involvement in future software selections or service enhancements.

Initially these services should be provided by the data warehouse implementation and monitoring team for a period up to three to six week months, or until the new data warehouse increment is established and relatively problem free. For enhancements or protection efforts, this level of support can be reduced to two months and one month respectively.

After establishing a Help and support Center function as part of the role out process, responsibility for this service can be passed to IT. With the migration of this service away from the core group of expertise, training and support of Help and support Center personnel must become part of the overall data warehouse training program. Due to the more volatile nature of this technology, support of Help and support Center personnel is critical to their ability to provide good service.

This process specifies how to collect, document, answer and/or escalate Help and supports, requests, and queries related to issues with the data warehousing environment. Problem documentation can be completed either by the Help and support Center representative and/or in conjunction with a form completed by the end user or IT support person requesting a service or action.

All inquiries, no matter how trivial should be logged, especially during the start of a new data warehouse or mart. These bits of information can form clues to taking proactive action to bigger problems before they emerge. Having a production ready data warehouse means support must be expedited in an efficient, responsive, and businesslike manner. At stake is the ability of the business to stay competitive if the business information the warehouse contains is not current, accurate, timely and available when needed.

This thought must be kept in mind by all Help and support Center personnel as they strive to answer those nagging questions: why queries don't run the way or as fast as they expect.

2.3.4 Management of data flow on network

Modern communication networks create large amounts of operational data, Including traffic and utilization statistics and alarm/fault data at various levels of detail. These massive collections of network-management data can grow the order of several Tera bytes per year, and typically Help and support hide "knowledge" that is crucial to some of the key tasks involved in effectively managing a communication network (e.g., capacity planning and traffic engineering). Besides providing easy access to people and data around the globe, modern communication networks also generate massive amounts of operational data throughout their life span.

As an example, Internet Service Providers (ISPs) continuously collect traffic and utilization information over their network to enable key network-management applications. If there is a mixed group of platforms for the data warehouse implementation, network management is going to be one of the most demanding responsibilities .

Not only are users coming constantly on-line, but users and equipment are invariably moving to new locations. The networking hardware is proliferating with LANs, WANs, hubs, routers, switches and multiplexers. Leaving behind all this is the next stage – users wanting to access internet based data sources along with the corporate data, requiring even greater bandwidth and network management resources. Managing this environment is one big challenge, capacity planning for the future is another. If the data warehouse team is not quite good in networking technology than there should be at least one person in the organization who understands technology.

If the data warehouse is implemented using a heterogeneous group of platforms, network management be one of the most difficult and tough tasks [26]. New users continuously come online and users along with equipment are invariably moving to new locations as well.

The networking hardware is always increasing in numbers with LANs, WANs, hubs, switches, routers and multiplexers. Users always want to access internet based data sources along with the corporate data, requiring even more bandwidth and network management resources. There should be some knowledgeable person in the organization who could handle these issues.

Some integrated tools are required to assist data warehouse team or the network management team in monitoring the network performance [17]. Fortunately there are several such tools now available, and enhancements are being made with each new release to improve their functionality. Because simple network management protocol (SNMP) is the common standard in this area; most vendors concentrate on SNMP based distributed network management features. Listed below are some of the features to look for in these tools [17]:

1. Distributed console support: The ability to access the tool from several different consoles.
2. Heterogeneous relational database support: The ability to work with many different database management systems, in a mixed environment.
3. Scalability: The tool should be able to work with an increasing number of servers and platforms without any loss in capability.
4. Interfaces for alerts (error messages requiring action): Should be able to support a variety of operating systems.
5. Security features: Should have features such as user ID authentication and auditing of attempted invalid accesses.
6. Tracking of network utilization in real time and in summary reports.

7. Measurement of network response time.
8. Graphical display of key measurements.
9. Troubleshooting assistance: The ability to provide diagnostics when problems occur.
10. Identification of under-utilized and over-utilized resources to permit load balancing and tuning.
11. Optimization tool to improve overall performance.

Even with these sophisticated tools, many warehousing systems find that their staff lacks the expertise to use them to full extent [17]. This is a complex area and if the staff members do not utilize the tools and associated methods frequently enough, they do not build up enough experience to become experts.

So some companies find it cost effective to use outside service providers who specialize in this area to help them identify their best options and sometimes, implement the recommendations. Such firms can supply network planning, design, implementation, management and monitoring services, either remotely or on site.

By proactively monitoring the network and resolving any bottleneck issues, it can analyze performance and put a plan in place to be able to support the critical applications [26]. There is no doubt that the new tools and services can provide much better insight into network performance than their predecessors. But it is still the job of network administrator to maintain network performance and meet the company's objectives. In developing network strategy consider the limitations of the current environment, as if it were not structured for the type of use to intending to place on it [30].

Using mechanisms like ODBC to move small amounts of data in batch for periodic updates is one thing while moving large amounts of data for both loading and querying on an ongoing basis requires careful planning.

The available network capacity has a huge impact on the data warehouse data management plan (data topology). It would be an ideal situation if the database management layer is defined before identifying the required network infrastructure. If a network infrastructure already exists, however, capacity planning must be completed before the data topology design for the data warehouse is finalized.

Consider the protocols available with the extraction and transformation software, database and information access software and check whether they are compatible or extendable. With the closing of the gap between technologies such as Asynchronous Transfer Mode (ATM) and Fiber Distributed Data Interface (FDDI), the distinction between the two greatly decreases. This increased compatibility and integration eases the planning of parallel based architectures with respect to data distribution and access, ranging over today's parallel server architecture, local and wide area networks.

These parallel architectures are transparent to the end user [30]. The knowledge worker accessing the data warehouse decision support system views the virtual parallel system through his or her personal computer system not caring if the access is local or distributed. His or her main consideration is and always remain based on performance, how much and how fast our architecture can provide the answers to a growing list of more complex and extensive business questions.

In developing a strategy for data staging and data replication use the DBMS's own data movement facilities to move data asynchronously between different database systems [30].Try to use application program controlled asynchronous data movement when data from multiple sources are required to insert aggregate data for better performance.

Data movement to multidimensional databases should rely on the data extract and movement facilities supplied by the multidimensional software

vendor. Along the same lines, unstructured data movement should be managed by the products vendor.

What role data replications play in data warehouse database management [30]? Data replication should be considered only when either

1. Data mirroring is required
2. Data distribution is required
3. Data movement of a subset of a data warehouse is required to update one or more dimensions of one or more marts that require this information.
4. Data replication should never be considered as a replacement for the data staging process by moving data directly from source systems to either an operational data store, data warehouse, and/or data mart.

2.3.5 Loading and Cleansing Data

One of the first issues companies need to confront is that they are going to spend a great deal of time loading and cleaning data. Some experts have said that the typical data warehouse project require companies to spend 85% of their time for it. While the percentage may or may not be as high as 85%, one thing to understand is that most vendors understate the amount of time needed to spend doing it. While cleaning the data can be complicated, extracting it can be even more challenging.

No matter how well a corporate prepares for the project management; they must face the scope of the project probably be broader than they estimate. While most projects begin with specific requirements, they conclude with data. Once the end users see what they can do with the data warehouse after it's completed, it is very likely that they place high demands on it. While there is

nothing wrong with this, it is best to find out what the users of the data warehouse need next rather than what they want right now.

Another matter that companies have to face is having problems with their systems placing information in the data warehouse. When a corporate enters this stage for the firstly, they find that problems that have been disappear for years suddenly appear. Once this occur, the business managers have to make the decision of whether or not the problem can be fixed via the transaction processing system or a data warehouse that is read only.

It should also be noted that a company often be responsible for storing data that has not be collected by the existing systems they have. This can be a headache for developers who run into the problem, and the only way to solve it is by storing data into the system. Many companies also find that some of their data is not being validated via the transaction processing programs.

In a situation like this, the data need to be validated. When data is placed in a warehouse, there are a number of inconsistencies that occur within fields. Many of these fields have information that is descriptive. When of the most common issues is when controls are not placed under the names of customers.

These cause headaches for the warehouse user that wants the data warehouse to carry out an ad hoc query for selecting the name of a specific customer. The developer of the data warehouse may find themselves having to alter the transaction processing systems. In addition to this, they may also be required to purchase certain forms of technology.

One of the most critical problems a company may face is a transaction processing system that feeds info into the data warehouse with little detail. This may occur frequently in a data warehouse that is tailored towards products or customers. Some developers may refer to this as being a granular issue. Regardless, it is a problem that wants to avoid at all costs. It is important to make sure that the information that is placed in the data warehouse is rich in detail

Loading and Cleansing Data is the core process of data integration and is typical Help and support associated with data warehousing. ETL which stands for extract, transform, and load is a three-stage process in database usage and **data warehousing**. It enables integration and analysis of the data stored in different databases and heterogeneous formats. After it is collected from multiple sources (extraction), the data is reformatted and cleansed for operational needs (transformation).

Finally, it is loaded into a target database, data warehouse or a data mart to be analyzed. Most of numerous extraction and transformation tools also enable loading of the data into the end target. It is a key process to bring all the data together in a standard, homogenous environment.

Its functions reshape the relevant data from the source systems into useful information to be stored in the data warehouse. Without these functions, there would be no strategic information in the data warehouse. If source data taken from various sources is not cleanse, extracted properly, transformed and integrated in the proper way, query process which is the backbone of the data warehouse could not happened.

ETL is a data combination function that involves extracting data from external sources (operational systems), alter it to fit business requirements, and ultimately loading it into a data warehouse To solve the problem, companies use extract, transform and load (ETL) technology, which includes reading data from its source, cleaning it up and formatting it uniformly, and then writing it to the target repository to be exploited .

Role and Responsibility Loading and Cleansing Data.

According to the function of data acquisition in corporate information factory in which the data warehouse and operational data store are populated from operational sources is the most technically Help and support challenging and difficult part of any data warehousing environment.

According to some industry experts approximately 60-85 percent of a data warehousing project effort is spent on this process alone. In today's high volume, client/server environment data acquisition techniques have to coordinate staging operations, filtering, data hygiene routines, data transformation and data load techniques in addition to cooperating with network technology to populate the data warehouse and operational data stores.

Taking the time to properly architect a highly integrated set of processes and procedures up front is the fastest way to achieve a smoothly running system that is maintainable and sustainable over the long haul. To accomplish an efficient, scalable and maintainable process, the Loading and Cleansing architect must have the following roles and responsibilities.

The Loading and Cleansing architect should have a close eye on the needs and requirements of the organization. He/she must understand the overall operational environment and strategic performance requirements of the proposed system. The architect must interact with the source system operational and technical staff, the project database administrator (DBA) and the technical infrastructure architects to develop the most efficient method to extract source data, identify the proper set of indexes for the sources, architect the staging platform, design intermediate databases needed for efficient data transformation and produce the programming infrastructure for a successful Loading and Cleansing operation.

To give a general idea of the functionality of these tools we mention their most prominent tasks, which include:

- The identification of relevant information at the source side;
- The extraction of this information;
- The customization and integration of the information coming from multiple sources into a common format;
- The cleaning of the resulting data set, on the basis of database and business rules,
- The propagation of the data to the data warehouse and/or data marts. To probe into the aforementioned issues,

The lifecycle of a data warehouse begins with an initial Reverse Engineering and Requirements Collection phase where the data sources are analyzed in order to comprehend their structure and contents. At the same time, any requirements from the part of the users (normally a few power users) are also collected. The deliverable of this stage is a conceptual model for the data stores and the activities.

In a second stage, namely the Logical Design of the warehouse, the logical schema for the warehouse and the activities is constructed.

Third, the logical design of the schema and processes is refined to the choice of specific physical structures in the warehouse (e.g., indexes) and environment-specific execution parameters for the operational processes. This stage is help and supported Tuning and its deliverable, the physical model of the environment.

In a fourth stage, Software Construction, the software is constructed, tested, evaluated and a first version of the warehouse is deployed. This process is guided through specific software metrics. Then, the cycle starts again, since data sources, user requirements and the data warehouse state are under continuous evolution.

An extra feature that comes into the scene after the deployment of the warehouse is the Administration task, which also needs specific metrics for the maintenance and monitoring of the data warehouse.

Loading and Cleansing is one of the most important sets of processes for the provisions and protection of Business Intelligence architecture and strategy . Time and thought are required to ensure the best architecture for its various components as well as for the selection of appropriate software tools and procedures within each component.

Ongoing Business Intelligence development demands a flexible, scalable and easily maintainable environment that can only come from an architected approach. , the information contained in a warehouse flows from the same

operational systems that could not be directly used to provide strategic information. Loading and cleansing functions reshape the relevant data from the source systems into useful information to be stored in the data warehouse. Without these functions, there would be no strategic information in the data warehouse.

If source data taken from various sources is not cleanse, extracted properly, transformed and integrated in the proper way, the extracted data often be stored in a central staging area where it cleanse and otherwise transformed before loading into the warehouse. An alternative approach to information integration is that of mediation: data is extracted from original data sources on demand when a query is posed, with transformation to produce a query result.
Once the initial data has been transferred to the data warehouse, the process must be repeated consistently. Data acquisition is continues process, and the goal of a company is to make sure the warehouse is updated on a regular basis.

When the warehouse is updated, it is often hard to determine which information in the source has changed since the previous update. The process of dealing with this issue is called changed data capture. This process has become a separate field, and there are a number of products currently are sold to deal with it.

It is important for data to be cleaned before it can be placed in the warehouse. The data cleansing process is usually done during the data acquisition phase. Any data that is placed in a warehouse before being clean pose a danger to the system, and it cannot be used.

The reason for this is because the data may not be correct if it is not cleaned, and a company may make incorrect decisions based on it. This could lead to a number of problems. For example, all the information within a data warehouse that means the same thing must be stored in the same form.

If there is information that reads "MS" and "Microsoft," even though they mean the same thing, only one of them can be used to recognize the element within the data warehouse.

2.3.7 Software and hardware

In order to build the first iteration of warehouse development it is necessary to have selected the technology on which to build the data warehouse. The selection of data warehouse technology - both hardware and software - depends on many factors, such as:

- The volume of data to be accommodated.
- The speed with which data is needed.
- The history of the organization.
- Which level of data is being built?
- How many users there are?
- What kind of analysis is to be performed?
- The cost of technology, etc.

Both hardware and software must be selected. The hardware is typically mainframe, parallel, or client/server hardware. According to W. H. Inmon the software that should be selected is for the essential data base management of the data as it exists on the hardware. In general the software is either full function data base management system or specialized data base software that has been

optimized for the data warehouse. A rough sizing of data requires to be done to decide the strength of the hardware and software policy. If the hardware and software policy are either much too big or are much too little for the amount of data that reside in the data warehouse, then no iterative development should occur until the fit is properly made.

If the hardware and data base management system software are much too large for the data warehouse, the costs of building and running the data warehouse be very expensive. Even though performance be no problem, development and operational costs and assets be a problem. On the other hand, if the hardware and data base management system software are much too little for the size of the data warehouse, then performance of operations and the finally end user satisfaction with the data warehouse suffer. At the outset it is important that there be a comfortable fit between the data warehouse and the hardware and data base management system software that house and manipulate the warehouse.

In order to conclude what the fit is like, the data needs to be sized. The sizing does not require being exact. If something, the sizing needs to err on the size of being too large, rather than too small. But a rough sizing of the data to be housed in the data warehouse at this point can save much grief at a later point in time if in fact there is not a comfortable fit between the data warehouse data and the environment it is built in. The estimate of the data warehouse data should be in terms of order of magnitude.

Of course, one of the issues that relates to the volume of data in the data warehouse is that of the level of granularity of data. If too much data is likely to be built into the data warehouse, the level of granularity can always be adjusted

so that the physical volume of data is reduced. Other software that needs to be considered is the interface software that provides transformation and metadata capability such as PRISM Solutions Warehouse Manager.

Over the last decade, the largest data warehouses have increased from 5 to 100 terabytes, according to Winter Corp., and by 2010, most of today's data warehouses be 10 times larger, according to The Data Warehouse Institute (TDWI). As data warehouses grow in size to accommodate regulatory requirements and competitive pressures, ensuring adequate database performance are a big challenge in order to answer more ad hoc queries from more people. When looking at the scalability, agility and readiness criteria more closely, it is clear that these criteria are implying both software and hardware functionality. There are limits to the performance of any individual processor (CPU) or individual disk. Hence, all high-performance computers include multiple CPUs and multiple disks. Similarly, a high-performance DBMS must take advantage of multiple disks and multiple CPUs.

Three infrastructure criteria, smart – or brainy – software and brawny hardware should not be separated in diagnosing infrastructure constraints. For example, the storage system should be able to read data back fast, but not at the expense of the security of payroll or other sensitive data. If see bottlenecks in current system, make sure to place requirements on both the software and hardware. Neither one of them by themselves can solve all bottlenecks.

Bottlenecks exist in both software and hardware. Simply stated, either software is not brainy enough, or – an unfortunately the hardware is not brawny enough. As the market-leading database for both OLTP and Data Warehouse applications, delivers all the smarts need and then some. Be sure to use these

smarts in the systems: ensure data security with Virtual Private Databases, achieve scalability with Real Application Clusters, Partitioning and Compression, and leverage the expertise of database administrators by standardizing on any type of application.

A system is balanced when the storage sub-system is capable of reading, writing and moving through the entire storage fabric – enough data to the database servers to have the CPUs adequately loaded. In other words, neither the IO capacity across the network, nor the bandwidth within the storage subsystem, or the CPUs should be a constraint to the system.

The Results are gaps in available technology and software, leaving users frustrated and their needs unmet. To overcome these problems warehouses needed to get their software and hardware updated in a timely manner to avoid any shortcomings in performance. Three strategies are available to make changes to this technical layer depending upon the scope, timeframe and criticality of the data warehouse environment. These strategies include [281]:

1. Installing new software releases, patches, hardware components or upgrades, and network connections (logical and physical) directly in the production environment.
2. Installing new software versions, hardware upgrades, and network improvement tasks in a temporary test environment and migrates or reconnects to production once certification testing has concluded.
3. Installing technical infrastructure changes into a permanent test or maintenance environment and migrate the production environment once certification testing has concluded.

2.3.8 Materialized View

A view can be materialized by storing the tuples of the view in the database. Index structures can be built on the materialized view. Consequently, database accesses to the materialized view can be much faster than recompiling the view. A materialized view is thus like a cache {a copy of the data that can be accessed quickly.} a materialized view provides fast access to data; the speed difference may be critical in applications where the query rate is high and the views are complex so that it is not possible to recomputed the view for every query. Materialized views are useful in new applications such as data warehousing, replication servers, chronicle or data recording systems, data visualization, and mobile systems. Integrity constraint checking and query optimization can also benefit from materialized views. A database that collects and stores data from several databases is often described as a data warehouse.

Materialized views provide a framework within which to collect information into the warehouse from several databases without copying each database in the warehouse. Queries on the warehouse can then be answered using the materialized views without accessing the remote databases. Provisioning, or changes, still occurs on the remote databases, and are transmitted to the warehouse as a set of modifications. Incremental view maintenance techniques can be used to maintain the materialized views in response to these modifications. While the materialized views are available for view maintenance, access to the remote databases may be restricted or expensive. Self-Maintainable views are thus useful to maintain a data warehouse.

A real-time data warehouse serves the purpose of monitoring the status of real-world objects in a dynamic environment. It enables enterprise managers to take effective decisions on the basis of the information that it provides them. Therefore, accuracy and timeliness of the information is a prime requirement.
In contrast to traditional data warehouses which perform batch-job data maintenance on a intervallic basis, in real-time data warehouses, updates are continuously supplied to the system, thus requiring the data to be refreshed to meet the freshness requirement. If these updates are not applied at the right time, then result in transactions reading stale data, which would thus, present problems for the business organization.

However, it is not just enough to apply updates; even business transactions must be allowed to read the updated data at the same time. But in a real-time system, updates and transactions come at a high speed which results in the need of an efficient scheduling strategy such that they do not conflict with each other. Particularly, query results in real time data warehouses are expected on real-time data. Nevertheless, data may be refreshed while the query is being processed, so most real-time data warehouse systems allow queries to read some version of data that is valid from the time the query has started. Depending on the frequency of data updates, some systems even permit reading slightly stale data. This is a type of trading of data timeliness for transaction timeliness. In any case, traditional process of running periodic maintenance batches does not satisfy data freshness requirements of real-time data warehouses.

Hence, in real-time data warehouses, we need efficient scheduling policies to deal well with write-only updates propagated from sources, view maintenances caused by updates, and read-only OLAP transactions.

When transactions' deadlines are top priority, reserving multiple data versions to avoid conflict between OLAP transactions and maintenance activities is shown to be an efficient solution. However, the versions take up too much space, and the cost for selecting data versions may affect transaction execution time, causing them to miss their deadlines. Though transaction deadlines may also be met by sacrificing data freshness requirements, this should be done in a controlled manner, since letting transactions read too old data does not yield any business value. Therefore a good scheduling approach should reduce transactions missing deadlines, increase data freshness, and reduce version scanning costs.

The issues of building a real-time data warehouse can be divided into two complementary parts:

(a) Quickly and robustly transforming changes from operational data sources into data warehouse records,

(b) Updating the data warehouse using those records in a timely manner.

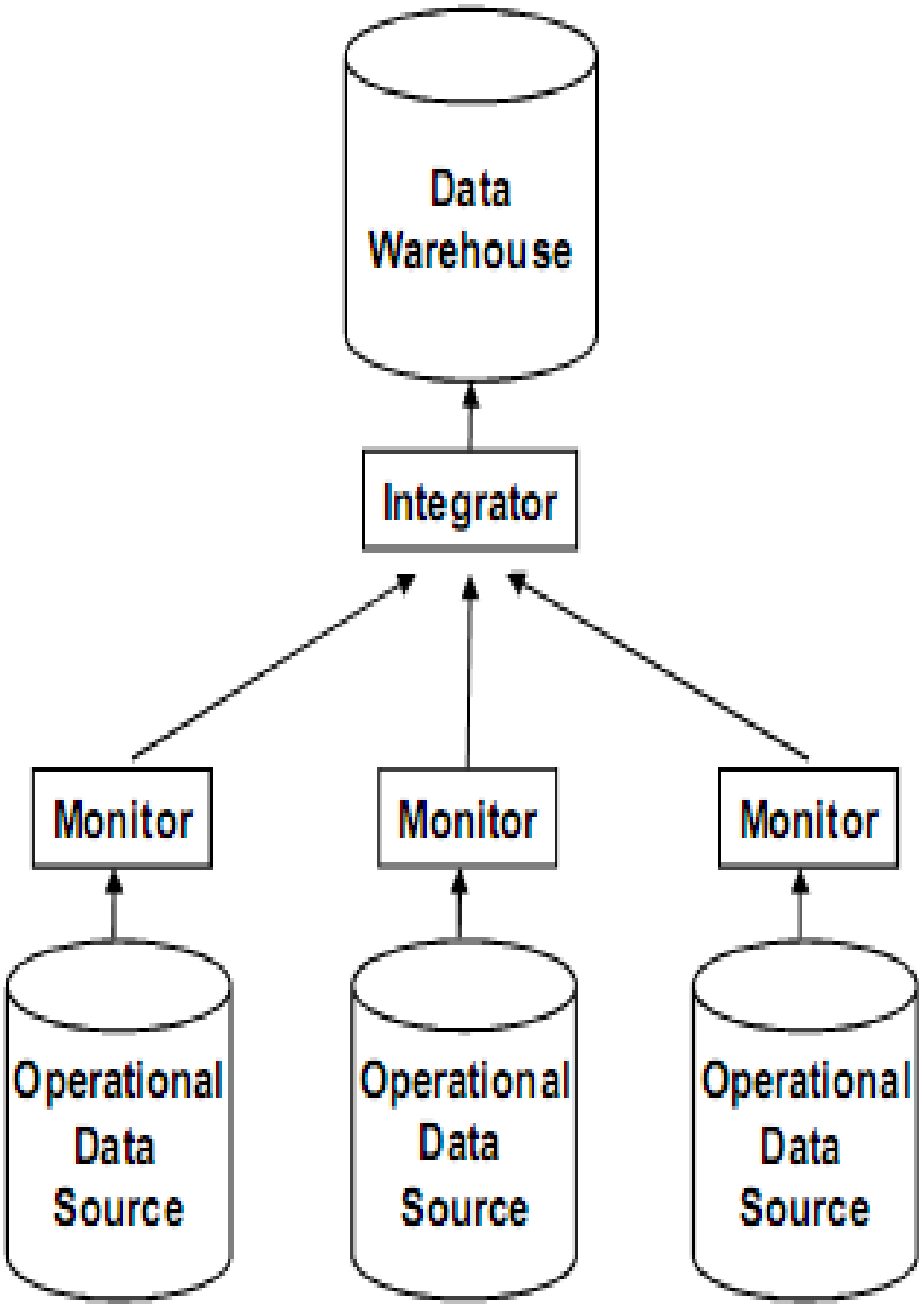

Figure2.2 - Monitor-Integrator Architecture.

The descriptions of such a method are as follows [see Figure]: A monitor resides on the top of each individual data source from which it collects changes of interest and propagate to the data warehouse. If the source supports relational-style triggers, the monitor only needs to pass on that information. Otherwise, it may need to extract the difference between successive snapshots of the source data Help and supported delta changes.

A centralized integrator at the data warehouse is responsible for translating delta updates from monitors to final records for loading purpose. In cases when the table to be updated in the data warehouse is constructed from more than one source (e.g., a join view), the integrator needs to build appropriate queries and send them to related sources.

The second problem is surprisingly identical to what techniques in the real-time database field pursue. In particular, they are also concerned with integrating updates to the database as soon as possible. The only difference lies in the availability of such updates: in real-time data warehouse, updates are normally obtained by joining with other data sources whereas in real-time database, they are assumed to be available at hand. Hence in some sense, much of what is known about real-time database scheduling, including theory and specific algorithms to testing procedure, can be applied in real-time data warehouse (obviously with some modifications). More specifically help and support, updates considered in ORAD are assumed to be in the format that can be used to update the data warehouse directly

While queries Help and supporting for up-to-date information are growing and the amount of data reflected to data warehouses has been

increasing, the time window available for making the warehouse up-to-date has been shrinking. Hence, an efficient view maintenance strategy is one of the outstanding issues in the data warehouse environment. This can improve the performance of query processing by minimizing OLAP queries down time and interference.

There can be roughly two different methods in reflecting data changes to materialized views: recompilation and incremental maintenance. Incrementally maintaining a materialized view includes computing and propagating only its changes. Compared to the sizes of base relations and views, their changes are generally very small. Hence, it is cheaper to compute only the changes of a view rather than to recomputed it from scratch.

Data warehouses usually contain a very large amount of data. In this scenario it is very important to answer queries efficiently therefore it need to use highly efficient access methods and query processing techniques. It is an important physical design decision to decide which indices to build and which views to materialize. The next major issue to deal with is how to use these indices and materialized views efficiently for maximum output. Optimization of complex queries is another important problem.

A data warehouse stores integrated information from multiple data sources in the form of materialized views over the source data. The data sources may be heterogeneous, distributed and autonomous. When the data is changes by any source, the materialized views at the data warehouse need to be updated accordingly. The process of updating a materialized view in response to the changes in the underlying source data is help and supported view maintenance. The view maintenance problem has evoked great interest in the

past few years. This view maintenance in such a distributed environment gives rise to inconsistencies since there is a finite unpredictable amount of time required for propagating changes from the data sources to the data warehouse and computing view updates in response to these changes.

Data warehousing is used for reducing the load of on-line transactional systems by extracting and storing the data needed for analytical purposes (e.g., decision support, data mining) [1]. A materialized view of the system is kept at a site Help and supported the data warehouse, and user queries are processed using this view.

View Maintenance Policies

A view maintenance policy is a decision about when a view is refreshed, independent of whether the refresh is incremental or not. The two common strategies used are:

Direct View Maintenance

A view can also be refreshed within the same transaction that updates the underlying tables. This view maintenance technique is help and supported immediate view maintenance. In this scenario the update transaction is slowed by the refresh step, and the impact of refresh increases with the number of materialized views that depend on the updated table.

Indirect View Maintenance

As an alternative to immediate view maintenance in this technique updates to the base tables are captured in a log and applied at a later stage to the materialized views. There are further three techniques in deferred view maintenance which are:

- Lazy: The materialized view is updated when a query accesses the view.
- Periodic**:** The materialized view is updated after a certain period of time Forced**:**
- The materialized view is refreshed after a certain number of changes have been made to the underlying tables.

It's the responsibility of the data warehouse team to make decision regarding the view maintenance strategies. A materialized view eliminates the overhead associated with expensive joins and aggregations for a large or important class of queries. Materialized views are of three types Materialized Views with Aggregates, Materialized Views Containing Only Joins and Nested Materialized Views. Maintaining a view is one of the most important tasks in warehousing environment. A materialized view eliminates the overhead associated with expensive joins and aggregations for a large or important class of queries.

Materialized views are of three types Materialized Views with Aggregates, Materialized Views Containing Only Joins and Nested Materialized Views. Maintaining a view is one of the most important tasks in warehousing environment. Nathan Folkert et al. have revealed the updates to the base table using bulk and partition operations, they refresh optimizer in the presence of partitioned table and materialized views, many recognize dependencies between

base tables and the Materialized on of very efficient refresh expressions. This makes Database Server more manageable and user friendly.

2.4 Proposed Model

2.4.1 Query Performance

In my proposed model I have combined records using grouping similar data concept to improve Response time and Searching time/Scan time. A *group* is a collection of data which are *similar* between them and are *dissimilar* to the data belonging to other group. Now we will define all the steps involved in process of grouping i.e my proposed model.

We will start with N groups containing similar records. Then we will make a symmetric matrix of N*N dimensions which will contain distance between each record with all other records. Distance will be measured on some pre defined structure. Now process will search for the pair of record with least distance that will be denoted by d_{uv}. Merge u and v group together and make a new group. This process will go on till we get a single group.

Algorithm for Group Model:

1. Start with N groups, and a single sample indicates one group.
2. Find the closest (most similar) groups and merge them into a single group, so that it has one group less.
3. Compute distances (similarities) between the new group and each of the old group.
4. Repeat steps 2 and 3 until all items group came into a single group of size N.

.

We have a table of database in which three fields phone no, customer name and sub locality.

Phone No	Customer Name	Sub Locality
22398	Amit	Pawanpuri
22245	Mukesh	Gandhi Colony
22268	Anil	Gandhi Colony
22562	Mohan	Karni nagar
22593	Amol	Karni nagar
22253	Rohit	Gandhi Colony
22264	Anita	Gandhi Colony

Table 2.1 – Records for proposed model

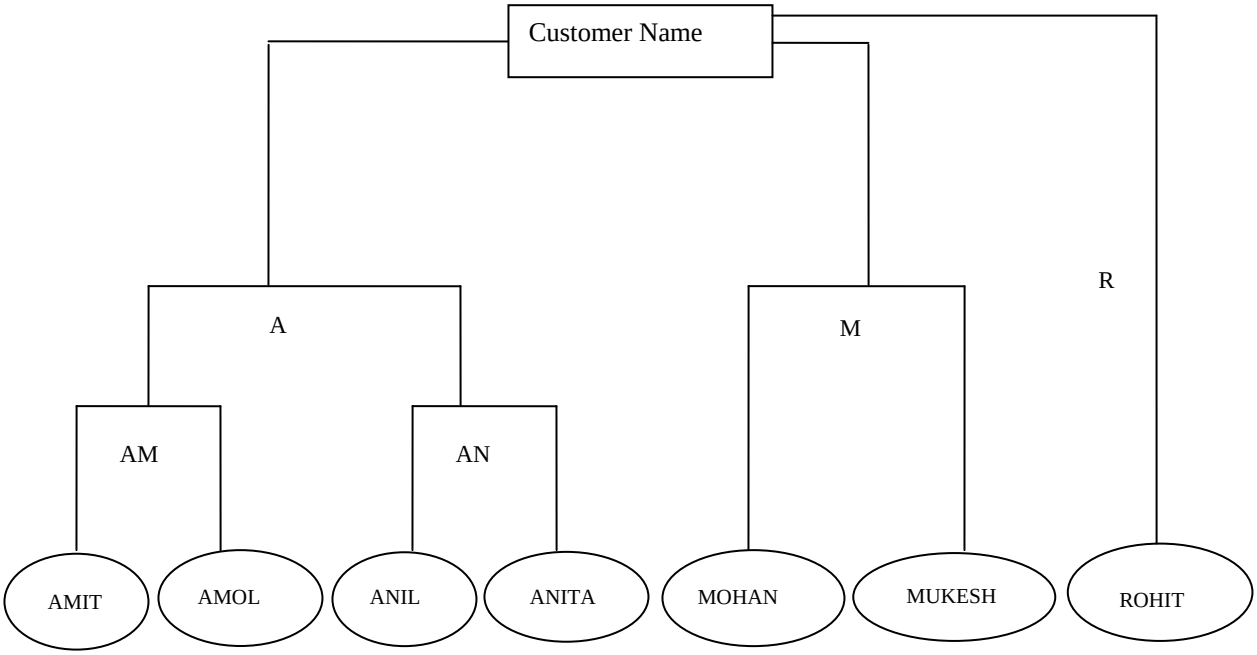

Figure – 2.3 Basic Structure of proposed model for alphanumeric records

Now we will show it by an example which is as follows:

Here we consider the BSNL dataset and especially the phone number, customer name and sub locality. In fist level we have taken exactly 26 alphabetic characters which start from A and end to Z. In second level of grouping we are telling to the system to form the group of data of same initial character with minimum distance. This method of grouping is repeated until data with minimum distanced are grouped in one group

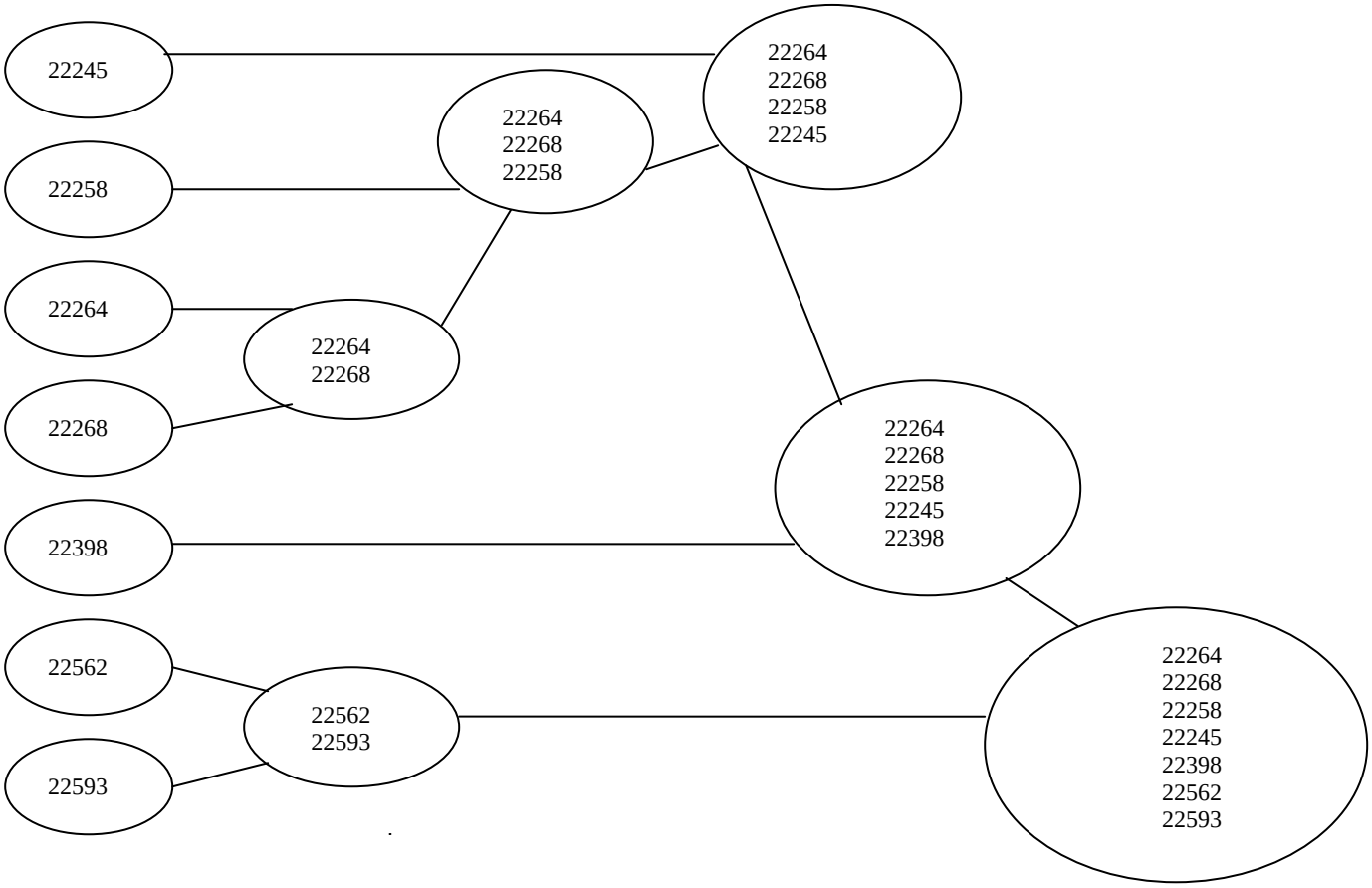

Figure – 2.4 Basic Structure of proposed model for numeric record

In fist level we have taken exactly number which start from 0 and end to 9. In second level of grouping we are telling to the system to form the group of data of same initial number with minimum distance. This method of grouping is repeated until data with minimum distanced are grouped in one group.

2.4.2 Coordination and Communication

The communication and coordination between processes also continues along with the training program. The communication and coordination between processes keeps the company users and IT users in contact with each other to have exchange of views, suggestions and any guidance towards enhanced performance of a data warehouse.

After the implementation of data warehouse one needs to address the issues of establishing and maintaining ongoing communication and training, providing help support services, and managing technical infrastructure updates concerning updated versions of hardware, software and services. Communication and training are two interrelated activities. A communication and training program is helpful in keeping the business community and IT community within the organization informed on the current and proposed future developments in the data warehousing environment.

A communication process also offers the data warehouse team to measure progress and identify and resolve issues before they become a serious problem. The communications program provides the business components with increased capabilities and functions. Training expands the communication process by maintaining a level competence in both the business and IT community as to the tools and mechanisms of the data warehouses.

As data warehousing starts to penetrate large numbers of enterprises, and it is currently doing so, it becomes critical that there be sufficient training assets to meet the demand produce by its wide spread adoption and dissemination. Further organizations and individuals in need of instruction should have some level of assurance that the training they get is the right training and they can benefit from the warehouse after the training.

2.4.3 Education, Training and documentation

The training program gives the users of data warehouse an insight into the qualities and capabilities of a data warehouse and teaches them the methods to benefit from it. Often the data warehouse projects fail because the users don't know how to use it according to the company needs. No one is going to use the data warehouse until they know how to use it, especially the company users who are more comfortable in receiving reports in a paper form instead of using computers for this purpose. Data warehouse help and support team members can continue the communication and coordination by keeping themselves in close contact with the company users and exchanging views on the present and proposed performance of data warehouse etc.

Training of data warehouse users is significant and provides the desired output. In most computing projects, management identifies the need for training, but does not always fund training. With every new database there is a need a training course, complete with reference materials. Every enhancement or change to the warehouse must be documented and communicated to warehouse users. The data administration department assumed responsibility for training and documentation of the data warehouse. While training users is essential, it distracts from future warehouse development unless new resources are allocated.

2.4.3 Help and support

While training reduces the number of data warehouse questions, a support infrastructure is the key to handling other support needs. In order to ensure success one needs to develop a support structure and plan an approach. When people are using the system, the questions will flow . If there are no questions than it is likely that no one is using the system. The question asked could be about the validity of the data itself, how to use the data, what calculations make sense and what levels of aggregation are valid, how to pick a report, how report measures were calculated, how to use the applications, how to change an application, and how to build an own application etc.

User support is crucial immediately following the deployment in order to ensure that the business community gets hooked . For the first several weeks following user education, the support team should be working proactively with the users. It can't sit back and assume that no news from the community is good news. If there is nothing heard from the business users it means that no one is using the data warehouse. In such a case the support professionals should turn up to the business community so that the users of the data warehouse have easy access to support resources. If problems with the data or applications are uncovered, immediately try to rectify the problems. Again if the warehouse deliverables are not of high quality, the unanticipated support demands for data reconciliation and application rework can be devastating.

2.4.4 Management of data flow on network

Management of data flow on network also plays its part in improving data warehouse performance. The network monitoring personals can use some specialized tools for monitoring network performance. Another approach for Management of data flow on network could be using the services of some specialized company in this field. Such firms can apply network planning, design, implementation, management and monitoring services either remotely or on site. Future planning for hardware and software resources for the data warehouse is compulsory for taking maximum output from it. In the beginning as the data warehouse usage is less, fewer resources are required, but as time passes and data warehouse starts becoming popular within the users, more hardware and software resources are required for a smooth running.

If there is a heterogeneous group of platforms for the data warehouse implementation, network data traffic is going to be one of the most demanding tasks . Not only are users coming constantly on-line, but users and equipment are invariably moving to new locations. The networking hardware is proliferating with LANs, WANs, hubs, routers, switches and multiplexers. Leaving behind all this is the next stage – users wanting to access internet based data sources along with the corporate data, requiring even greater bandwidth and network management resources. Managing this environment is one big challenge; capacity planning for the future is another. If the data warehouse team is not quite good in networking technology than there should be at least one person in the organization who understands technology.

2.4.5 Software and Hardware

Capacity planning refers to determining the required future configuration of hardware and software for a network or data ware house. There are numerous capacity planning tools in the market used to monitor and analyze the performance of the current hardware and software. The capacity planning enables the determination of sufficient resources so that user satisfaction can be maximized through timely, efficient and accurate responses.

Capacity planning is important when starting a new organization, extending the operations of an existing business, considering additions or modifications to product lines, and introducing new techniques, equipment and materials. In business, capacity is the maximum rate of output for a process. This means that capacity is the work that the system is capable of doing in a given period of time. The goal of capacity planning is to meet current and future demand with a minimal amount of waste.

2.4.6 Loading and Cleansing Data

Loading and Cleansing Data is the core process of data integration and is typical Help and support associated with data warehousing. It enables integration and analysis of the data stored in different databases and heterogeneous formats. After it is collected from multiple sources (extraction), the data is reformatted and cleansed for operational needs (transformation). Finally, it is loaded into a target database, data warehouse or a data mart to be analyzed. To solve the problem, companies use extract, transform and load (ETL) technology, which includes reading data from its source, cleaning it up and formatting it uniformly, and then writing it to the target repository to be exploited . It is important for data to be cleaned before it can be placed in the

warehouse. The data cleansing process is usually done during the data acquisition phase. Any data that is placed in a warehouse before being clean will pose a danger to the system, and it cannot be used.

To load a data warehouse, regular loading or propagation of data from operational systems is needed. A schedule for summarizing, loading, and making the information available to the user community needs to be developed and it should be presented to the user community.

For e.g. daily summary data may be available by 7 AM the next morning and weekly summary data by 8 AM Monday morning. The users should also know if and when the data was loaded.

It is also necessary to develop procedures for managing the results of a bad load. For e.g. there should be some defined procedures if the data loaded is corrupted due to some operational mistakes and the problem with data is discovered after some time. In that case the data needs to be reloaded. One needs to consider what are the impacts of data reloading, how it will be reloaded, what will happen if data is reloaded during peak hours (any effects on operational systems)? User notification regarding corrupted data should be part of the data loading procedure.

Factors affecting data loading performance include (30):

1. The decreasing batch window (time available for an intensive batch processing operation such as a disk backup) to load data from more and more sources, coupled with increased usage of the data warehouse.
2. The frequency and size of these loads.
3. The changing natures of these loads as source systems change or are replaced.

4. The increasing demand for more metadata regarding the data to be loaded.
5. If load performance remains an issue, consider maintaining a synchronous replica of the full database to source data and to the warehouse.

All these problems mean that the data warehouse team must plan for and examine performance on an ongoing basis. Creating a performance management system that logs all relevant data or selecting a product that provides this functionality will arm you with a critical tool in the fight to keep on top of the growing data management problem.

Do not consider loading data based on real time source to target mapping. Data quality is hard to monitor with this method, and critical performance problems do occur that affect the source systems and the data warehouse. Data can be loaded efficiently by dropping all indexes and rebuilding them after the load completes.

Turning off row level locking if possible can help in achieving better performance. DBMS's bulk loading facilities can be used to load data as well if available. Try to turn off DBMS journaling (tracking data changes). Data load personnel should ensure that the data staging tables map directly to the data warehouse data mart tables. These personnel should also try to calculate and prepare derived data for loading into data warehouse. Cleaning and transforming the data in the data staging environment prior to loading can be a great help in improved data loading performance.

Data warehouse maintenance staff can plan the division or segmenting of the big dimensions of warehouse, such as customer and product, by subtype,

defining each to separate physical tables prior to mapping the data across the disk drives. The parallel loading features of the processors can be used to further enhance the data loading performance. Scaling up the bandwidth of the network to accommodate more traffic for e.g. assuming 15 GB of change data per day at

1. 1 MB per second will take 41 hours to load
2. 10 MB per second will take 25 minutes to load
3. 100 MB per second will take 2.5 minutes to load

2.4.7 Materialized view

A basic requirement for the success of a data warehouse is the ability to provide decision makers with both accurate and timely consolidated information as well as fast query response times. For this purpose, a common method that is used in practice for providing higher information and best response time is the concept of materialized views, where a query is more quickly answered. One of the most important decisions in designing data Warehouse is selecting views to materialize for the purpose of efficiently supporting the decision making. The view selection problem defined is to select a set of derived views to materialize that minimizes the sum of total query response time & maintenance of the selected views. So the goal is to select an appropriate set of views that minimizes total query response time and also maintains the selected views.

The selection of views to materialize is one of the most important issues in designing a data warehouse. So as to achieve the best combination of good query response where query processing view maintenance cost should be

minimized in a given storage space constraints. The proposed algorithms are found efficient as compared to other materialized view selection and maintenance strategies. The total cost, composed of different query patterns and frequencies, were evaluated for three different view materialization strategies: 1) all-virtual-views method, 2) all materialized-views method, and 3) **proposed materialized-views** method. The total cost evaluated from using the ***proposed materialized-views*** method was proved to be the smallest among the three strategies. Further, an experiment was conducted to record different execution times of the proposed strategy in the computation of a fixed number of queries and maintenance processes. Again, the ***proposed materialized-views*** method requires the shortest total processing time.

Materialized view is a strategy used to provide fast answers to user queries. But it is important to have updated views whenever the base tables upon which views are built are updated. It is the responsibility of data warehouse help and support team to devise a flexible and optimal strategy for maintenance of materialized views.