Design Database Strategies on basis "GOMWTDS "Methods

Shailesh R. Thakare,
Professor,
Department of Computer Science & Engineering,
P.R.Pote(Patil) College of Engg & Mgmt, Amravati
(Maharastra), India

C.A. Dhawale
Professor,
Department of Computer Science & Engineering,
P.R.Pote(Patil) College of Engg & Mgmt, Amravati
(Maharastra), India

ABSTRACT

Database is not static but rapidly grows in size. These issues include how to allocate data, communication of the system, the coordination among the individual system, distributed transition control and query processing, concurrency control over distributed relation, design of global user interface, design of component system in different physical location, integration of existing database system security. Design Strategies of Database on basis "GOMWTDS " Methods.

1. INTRODUCTION

Top-down and bottom up approach are the two major design strategies for "GOMWTDS " Methods.. Although these two approaches carry out very different design process, the necessity of applying one approach to complement another is possible since real applications are likely to be too complicated to fit in just one approach. The problems of effectively partitioning a huge dataset and of efficiently alleviating too much computing for the processing of the partitioned data have been critical factor for scalability and performance. In today's data deluge the problems are becoming common and will become more common in near future. The principle "Make common case fast" (or "Amdahl's law" which is the quantification of the principle) can be applied to make the common case faster since the impact on making the common case faster may be higher, while the principle generally applies for the design of computer architecture.

1.1 Top-down approach

Top-down design process is mostly used in designing system from scratch. Figure1 illustrates the process of top-down design. The process starts from a requirement analysis phase including analyzing of the company situation, defining problems and constraints, defining objectives, and designing scope and boundaries. The next two activities are conceptual design and view design. Focus on the data requirements, the conceptual design deals with entity relationship modeling and normalization. It creates the abstract data structure to represent the real world items. The view design defines the user interfaces. The conceptual schema is a virtual view of all databases taken together in a distributed database environment. It should cover the entity and relationship requirement for all user views. Furthermore, the conceptual model should support existing applications as well as future applications. The definition of the global conceptual schema (GCS) comes from the conceptual design. The next step is distribution design. The global conceptual schema and the access information collected from the view design activity are inputs of this step. By fragmenting and distributing entities over the system, this step designs the local conceptual schemas. Therefore, this step can be further divided into two steps: fragmentation and allocation. Distribution design also includes the selection of DBMS software in each site. The mapping of the local conceptual schemas to the physical storage devices is accomplished through the physical design activity. Throughout the design and development of the distributed database system, constant monitoring and periodic adjustment and tuning are also critical activities in order to achieve successful database implementation and suitable user interfaces.

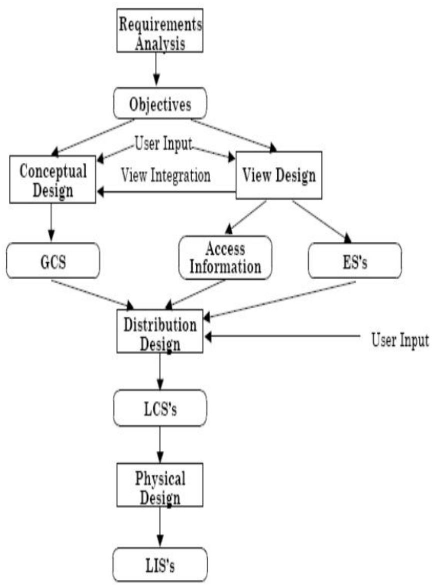

Figure 1: Top-down Design Process

1.2 Bottom-up approach

Bottom-up approach is suitable when the objective of the design is to integrate existing database systems. The bottom-up design starts from the individual local conceptual schemas and the objective of the process is integrating local schemas into the global conceptual schema. One of the most important aspects of design strategy is to determine how to integrate multiple database system together. Implementation alternatives are classified according to the autonomy, distribution, and heterogeneity of the local systems.

Autonomy indicates the independency of individual DBMS. In the autonomous system, the individual DBMS are able to perform local operations independently and have no reliance on centralized service or control. The consistency of the whole system should not be affected by the behavior of the individual DBMS. Three possible degrees of autonomy are tight integration, semiautonomous system, and total isolation. In a system which is tightly integrated, although information is stored in multiple databases, users only see a single image of the entire system. One of the DBMS controls the processing of the user request. The DBMS in semiautonomous system can operate separately and they are also willing to share their local data. In total isolated systems, individual DBMS do not know the existence of other DBMS.

The physical distribution of data over multiple sites is another characteristic of distributed databases. Distributed system can be classified as client/server distribution or peer-to-distribution based on how the data are distributed and how to manage them. Heterogeneous DDBMS integrate multiple independent databases into a single distributed database system and provide transparency of the heterogeneity. Individual DBMS can implement different data model, use different query language and transaction management protocols. Moving along the distribution dimension, the client/server distribution is introduced when the system is distributed with an integrated view providing to users. This implementation requires assigning the control of the entire system to one DBMS (single server) or several DBMS (multiple servers). The server(s) control each user request although the request might be serviced by more than one DBMS. The other scenario is that the system is fully distributed, but the distribution is transparent to the user. Each DBMS provides the identical functionality and there is no distinction among clients and servers. In distributed system, external schemas are defined as being above a global conceptual schema (GCS) which describe the logical data structure of the entire system. The global conceptual schema provides distribution transparency to users. It is the union of local conceptual schemas of local database systems which describe the logical organization of data at each site. The physical data organization on each site in the system is presented by local internal schema

2. "GOMWTDS " Methods:- These Methods are follows

2.1 Design Goal-oriented Schema :-

Goal-oriented requirements analysis starts high-level goals, which are refined and interrelated to produce a goal model. The goal model captures not a single, but several alternative sets of data requirements, from which a particular one is chosen to generate the conceptual schema for the database-to-be.

2.2 Design OMT (Object Modeling Technique):-

The OMT (Object Modeling Technique) method because it integrates a minimal set of concepts shared by several methods. These concepts are sufficient for our project. Moreover, OMT is based on various standards (Entity/ Association, Data flow diagram, State transition diagram) extended to the object paradigm. At last, the OMT graphical representation is expressive enough. A global conceptual schema of a DB managing a research centre documentation.

2.3 Database Designer's Workbench :-

The Database Designer's Workbench (or "Workbench") is a graphics oriented decision-support system to assist with the design of all aspects of a computerized database, from the initial specification of the system's requirements through its final physical structure. It provides a wide variety of design aids, or "tools," for designers to explore many design alternatives and to evaluate them precisely. These tools are presented in a homogeneous, graphically oriented environment that allows a designer to use familiar representations, store incomplete designs, progress smoothly from one design phase to the next, and iterate over previous design stages. The Workbench is therefore a real asset to the database design practitioner, seeking to improve productivity and the quality of design.

2.4 Tree-based Database Design

A database design approach based on tree structure is proposed. Tree structure is in common use and can express the level relation between or among object compensation items adequately the attributes of entities, and relationships between entities are made.

2.5 Database Design Based on DFDC :-

Standard database design consists of i) requirements analysis (resulting in a conceptual schema for the data to be stored, using a notation such as ER or UML Class Diagram), ii) logical design (resulting in a relational database schema in SQL), and iii) physical design (resulting in optimized access structures) DFDC consists of tables and directed lines with formula

Mark defined as 2-triples. *DFDC= (T, L)*, *T*: collection of tables; *L*: collection of linesTable *T* is defined as 2-triples, *T= (N, I)* where *N* is the name of the table (see for example *T1* and *T2* in Fig. 1), and *I* is the set of the *item*.And an *item* is denoted to be *I= (IF, IN)* where *IF*: Keywords flag for items, valued for *K* or *FK*, *K* represents keywords, while *FK* can be taken in designing and final stage, representing foreign keys in database. *IN* includes item's name and value. A *Line* is denoted to be *L=(ST.Ii,TT.Ij,F)* where i) *ST.Ii* is the source item meaning that the start of this line is the *Ii* item in table ST, ii) *TT.Ij* is the target item meaning that the end of this line is the *Ij* item in table *TT*, and iii) F is a formula on the *line*.

2.6 Database Design on based B+Tree:-

B+tree is the tree which assists minimization of addresses to bulk storage at a necessary information retrieval balanced on a height. Every knot of B-tree, unlike a binary tree, can contain the different amount of the keys and references on descendants. The basic line of classic B-tree is homogeneity of his knots (every knot contains only the keys and references on descendants).

The chart of B+-tree allows to keep in bulk storage (tables of database) not only the keys but also records. A way from the top of B+-tree (root table of database) to the sheet table has identical length, but their structure has substantial differences. The internal tables of darabase keep only the keys (way) and references on daughters (detail) tables. A sheet table keeps all keys, present in a concrete B+-tree, and all records.

2.7 2.7DatabaseDesign on based Star Model:-

Design database, there are requirement centralized control of database as well as Integration of system at that time specified the star model.

Star model of database design is divided into three types: control data, integrated data, and security control.

- Control data: by controlling the data, integrated system must provide to create data tables and to delete data tables. This model can not bring the impact of additions and deletions from database.
- Integrated data: this model must be fit to get together data from different government departments
- Secure access: it is must be field-level security.

3. CONCLUSION

In the above proposed work, we design Strategies the database on the bases using various methods it is help to improve the performance as well as enhanced the quality of databases, and we can improve the clarity about the database.

4. REFERENCES

[1] Richard E. COBB ”The Databases Designergs Workbench” Information System Research Group.(1984)

[2] Dixu “Distributed Database System Design” Minnesota State University Kankato.

[3] Lei Jiang , Thodoros Topaloglou, Alex Borgida, john Mylopoulos “Goal Oriented Conceptual Designing” (2007)

[4] Min CAO, Shu LUO, Huaikou Miao”Tree Based database desiging” School of Computer Engineering & Science 4th International Confernec on computer Science & Education.

[5] Lubomyr Sikora, Yuliya miyushkovych “B+ tree in database Design for Decision making Information System” (2010).

[6] CHEN Bingchuan, LI Lei , Wang Heyong “New Methods of Database design Basd on DFDC” Second International Confernec on computer Science & Education (2010).

[7] Wang heyong “Star like model database design for GEIS” Second International Confernec on E business & E government”