International Journal of Engineering and Management Research, Vol. 2, Issue-4, August 2012
ISSN No.: 2250-0758
Pages: 11-14
www.ijemr.net

Hierarchical Model Leads To the Evolution of Relational Model

[1]Gaurav Jindal [2]Simmi Bali
[1]Assistant Professor, Gitarattan International Business School, Rohni, New Delhi
[2]Website Developer, Asiawebnet Pvt Ltd., Rohini, New Delhi
[1]gauravjindal05@gmail.com, [2]simmicareer07@yahoo.com

ABSTRACT
A hierarchical database model is a data model in which the data is organized into a tree-like structure. The structure allows representing information using parent/child relationships: each parent can have many children, but each child has only one parent (also known as a 1-to-many relationship). Redundancy would occur because hierarchical databases handle one-to-many relationships well but do not handle many-to-many relationships well. Faced with these serious problems, the computer brains of the world got together and came up with the network model. Network Database looks like a hierarchical Database in that you can see it as a type of tree. However, in the case of a Network Database, the look is more like several trees which share branches. Thus, children can have multiple parents and parents can have multiple children. Nevertheless, though it was a dramatic improvement, the network model was far from perfect. Most profoundly, the model was difficult to implement and maintain. Most implementations of the network model were used by computer programmers rather than real users. What was needed was a simple model which could be used by real end users to solve real problems and this lead to the evolution of Relational Model.

KEYWORDS:
Database, Database Model, Hierarchical Database, Network Database, Relational Database

I. INTRODUCTION

In a hierarchical database, the data is organized into a tree structure. The tree structure allows information to be repeated using a parent and child type relationship. Each parent item can have many child items; however each child item can have only one parent. Attributes for each record are stored within entity types, which are the equivalent of tables. Records are stored in rows while associated attributes are stored within columns. A common example of a hierarchical database is IBMs IMS database.

Network database structures were developed to allow retrieval of specific records. They allow any given record to point to any other record in the database. Networks solve the problem of having to backtrack all the way to a joining "branch" of the database. However, this wide range of possible connections is also the weakness of applying network structures to practical problems since it was just too complex to allow every record to point to every other record.

The breakthrough came from basic research conducted independently by C. J. Date and E. F. Codd using relational algebra. They were able to show that relational databases created out **of** a series of interrelated tables were, in fact, far more flexible and versatile than either the hierarchical or network database structures. Whereas the hierarchical and network database structures rely on physical relationships in the form of storage addresses, relational database structures use implicit relationships that can be implied from the data.

II. REVIEW OF LITERATURE

A **database model** is the theoretical foundation of a database and fundamentally determines in which manner data can be stored, organized, and manipulated in a database system. It thereby defines the infrastructure offered by a particular database system. The most popular example of a database model is the relational model. A data model is not just a way of structuring data: it also defines a set of operations that can be performed on the data. The relational model, for example, defines operations such as select(project) and join. Although these operations may not be explicit in a particular query language, they provide the foundation on which a query language is built.

The discipline of data modeling initially became established because it provided way for specifying the structures of data* in actual file systems followed by

database management systems (DBMSs). This led to the introduction of the network and the hierarchical models in the 1960s exemplified by the DBMSs called Integrated Data Store (IDS) of Honeywell (network model) and Information Management System (IMS) of IBM (hierarchical model) [1].

Data models provide a way in which the stored data is organized as specified structure or relation for quick access and efficient management. Many models, such as Hierarchical model, Network model, entity- Relationship model, Functional model, Relational model, Object-Oriented model, has come into existence and played important roles since the emergence of the database management systems [2].

III. IMPORTANCE OF THE STUDY

A data model [3] is a (relatively) simple abstraction of a complex real-world data environment. Database designers[4] use data models to communicate with applications programmers and end users. The basic data-modeling components are entities, attributes, relationships, and constraints. Business rules are used to identify and define the basic modeling components within a specific real-world environment.

DATA
Data is a collection of facts, such as values or measurements. It can be numbers, words, measurements, observations or even just descriptions of things.

DATABASE
A database is an organized collection of data, today typically in digital form. The data are typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies).

DATABASE MANAGEMENT SYSTEM
A database management system (DBMS) is a software package with computer programs that controls the creation, maintenance, and use of a database. It allows organizations to conveniently develop databases for various applications. A database is an integrated collection of data records, files, and other objects. A DBMS allows different user application programs to concurrently access the same database. DBMSs may use a variety of database models, such as the relational model or object model, to conveniently describe and support applications.

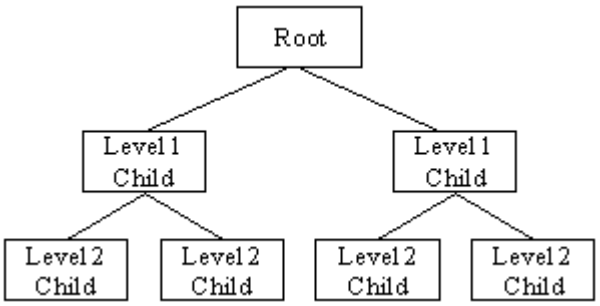
IV. OBJECTIVES

- Understand the hierarchy of data.
- Learn database structures and how they work.
- Learn how to relate tables together in a database.
- Recognize the difference between a database and a DBMS
- Understand the database concept.
- Learn methods for determining data needs.

V. RESULT & DISCUSSION

Hierarchical Database
Hierarchical databases are some of the oldest and simplest kinds of database. They arrange data in a "tree" structure, which is similar to folders and files on a computer. Just as a file on a computer sits in one folder, every record in the database has one "parent." Hierarchically arranged data is often described as having only parent/child relationships. Hierarchical Database Model defines hierarchically-arranged data.

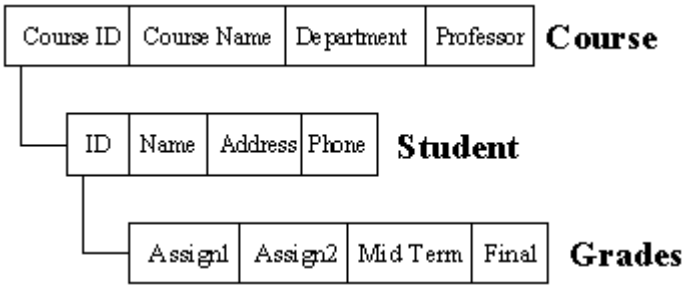
Perhaps the most intuitive way to visualize this type of relationship is by visualizing an upside down tree of data. In this tree, a single table acts as the "root" of the database from which other tables "branch" out. You will be instantly familiar with this relationship because that is how all windows-based directory management systems (like Windows Explorer) work these days. Relationships in such a system are thought of in terms of children and parents such that a child may only have one parent but a parent can have multiple children. Parents and children are tied together by links called "pointers" (perhaps physical addresses inside the file system). A parent will have a list of pointers to each of their children.



This child/parent rule assures that data is systematically accessible. To get to a low-level table, you start at the root and work your way down through the tree until you reach your target. Of course, as you might imagine, one problem with this system is that the user must know how the tree is structured in order to find anything!
If a change in the data is necessary, the change might only need to be processed once. Consider the student flatfile database example from our discussion of what databases are:

Name	Address	Course	Grade
Mr. Rohit	123 Rohini	Chemistry 102	C+
Mr. Rohit	123 Rohini	Mathematics	A
Mr. Rohit	122 Rohini	Data Structures	B
Mr. Rohit	123 Rohini	English 101	A
Ms. Garima	88 Pitampura.	Psychology 101	A
Mrs. Dimple	100 Shahdra	Psychology 102	A
Ms. Garima	88 Pitampura	Human Cultures	A
Ms. Garima	88 Pitampura	European Governments	A

If we implemented this in a hierarchical database model, we would get much less redundant data. Consider the following hierarchical database scheme:



Problem with Hierarchical Model

However, as you can imagine, the hierarchical database model has some serious problems. For one, you cannot add a record to a child table until it has already been incorporated into the parent table. This might be troublesome if, for example, you wanted to add a student to who had not yet signed up for any courses.

Worse, yet, the hierarchical database model still creates repetition of data within the database. You might imagine that in the database system shown above, there may be a higher level that includes multiple course. In this case, there could be redundancy because students would be enrolled in several courses and thus each "course tree" would have redundant student information.

Redundancy would occur because hierarchical databases handle one-to-many relationships well but do not handle many-to-many relationships well. This is because a child may only have one parent. However, in many cases you will want to have the child be related to more than one parent. For instance, the relationship between student and class is a "many-to-many". Not only can a student take many subjects but a subject may also be taken by many students. How would you model this relationship simply and efficiently using a hierarchical database? The answer is that you wouldn't.

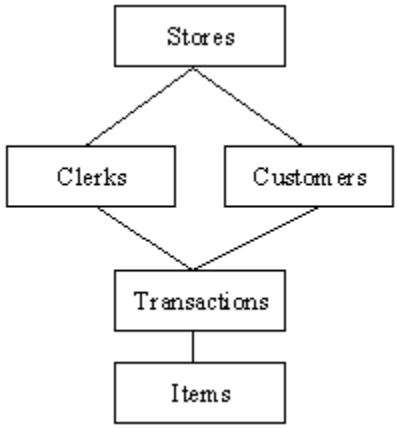
Though this problem can be solved with multiple databases creating logical links between children, the fix is very kludgy and awkward. Faced with these serious problems, the computer brains of the world got together and came up with the network model.

Network Database

In many ways, the Network Database model was designed to solve some of the more serious problems with the Hierarchical Database Model. Specifically, the Network model solves the problem of data redundancy by representing relationships in terms of sets rather than hierarchy.

The network model is very similar to the hierarchical model actually. In fact, the hierarchical model is a subset of the network model. However, instead of using a single-parent tree hierarchy, the network model uses set theory to provide a tree-like hierarchy with the exception that child tables were allowed to have more than one parent. This allowed the network model to support many-to-many relationships

Visually, a Network Database looks like a hierarchical Database in that you can see it as a type of tree. However, in the case of a Network Database, the look is more like several trees which share branches. Thus, children can have multiple parents and parents can have multiple children.



Problems with Network Model

Nevertheless, though it was a dramatic improvement, the network model was far from perfect. Most profoundly, the model was difficult to implement and maintain. Most implementations of the network model were used by computer programmers rather than real users. What was needed was a simple model which could be used by real end users to solve real problems. This problem leads to the evolution of Relational Model.

Relational Model

Relational databases have no problems with many-to-one or many-to-many relationships. Their records

are built as multiple "tables," rather than tree structures, and each record on a table has a unique identifier. A company could then have a table with the names of all the parents, a table with the names of all the children, and each record on the parent table could have a relationship with one (or more, or none) of the unique records on the child table--that relationship being "is the parent of." The ability to give records such relationships is what give relational databases their name.

Advantages of Relational Model

Relational databases prevent errors by allowing one record to apply to any number of other tables. A child record could be used in a "is the child of" relationship, and the same record could be referred to in a table of "children attending the company picnic." By preventing duplication, the same information can be used in many different ways, without accidentally altering a record.

Also, relational databases are very good for providing other kinds of data hidden in the records, using queries written in Structured Query Language, or SQL. This enables you to explore the database in ways not immediately apparent, such as finding all the children over a certain age, or all the parents with three or more children.

VI. CONCLUSION

A relational database is a wonderful piece of equipment for storing large quantities of data efficiently. In this tutorial I focused mainly on building the database model. These models can be implemented on any RDBMS and queried using the Structured Query Language

REFERENCES

[1] Navathe S. (1992), COMMUNICATIONS OF THE ACM, Vol.35, No.9

[2] Lin Caixue (2003), Object Oriented Database Systems: A Survey

[3] Michael R. McCaleb (1999). "A Conceptual Data Model of Datum Systems". National Institute of Standards and Technology.

[4] Jan L. Harrington (2000). Object-oriented Database Design Clearly Explained.

[5] Beynon-Davies P. (2004). Database Systems 3rd Edition. Palgrave, Basingstoke, UK. ISBN 1-4039-1601-2