Data manipulation techniques across databases especially handling of spatial data

Rucche Sharma*
Dr. Amit Gupta**

Abstract

The paper compares data manipulation techniques across various databases. It explores the improvement in managing different types of data including spatial data. The comparison also takes into account primitive database systems or file systems till the advanced Spatial databases. The paper highlights the fact that as concepts evolved across systems, front-end became easier and back-end took the bulk of task in manipulating data.

Keywords: Spatial Data, data manipulation, modeling spatial data, ORDBMS.

Introduction

Data are raw facts that constitute building blocks of information. Database is a collection of information and a means to manipulate data in a useful way, which must provide proper storage for large amounts of data, easy and fast access and facilitate the processing of data. Database Management System (DBMS) is a set of software that is used to define, store, manipulate and control the data in a database. From pre-stage flat-file system, to relational and object-relational systems, database technology has gone through several generations and has 40 years history.

Evolution

Ancient History
Data are not stored on disk; programmer defines both logical data structure and physical structure, such as storage structure, access methods, I/O modes etc. One data set per program: high data redundancy. There is no persistence; Random access memory (RAM) is expensive and limited, programmer productivity is low.

*Reseacrh Scholar, IGNOU, New Delhi, INDIA
**Reader, MAIMS, IP University, New Delhi, INDIA

Authors are grateful to the anonymous referees for comments and suggestions for improving the text and contents of the paper. Authors alone are responsible for any errors/mistakes still remaining in the paper.

File-Based
Predecessor of database, data is maintained in a flat file. Processing characteristics determined by common use of magnetic tape medium.

- Data are stored in files with interface between programs and files. Mapping happens between logical files and physical file, one file corresponds to one or several programs
- Various access methods exist, e.g., sequential, indexed, random
- Requires extensive programming in third-generation language such as COBOL, BASIC.
- Limitations:
  - Separation and isolation: Each program maintains its own set of data, users of one program may not aware of holding or blocking by other programs.
  - Duplication: Same data is held by different programs, thus, waste space and resources.
  - High maintenance cost such as ensuring data consistency and controlling access.
  - Sharing granularity is very coarse.
  - Weak security.
- Handling of spatial Data: No data type available, hence manipulation would happen through front-end. Programming had to be complex, which would thereby reduced processing speed. Also, due to no redundancy checks, integrity of data is also questioned.

Non-relational database
A database provides integrated and structured collection of stored operational data which can be used or shared by application systems. Prominent hierarchical database model was IBM's first DBMS called IMS. Prominent network database model was CODASYL DBTG model; IDMS was the most popular network DBMS.

Hierarchical data model
- Mid 1960s Rockwell partner with IBM to create information Management System (IMS), IMS DB/DC lead the mainframe database market in 70's and early 80's.
- Based on binary trees. Logically represented by an upside down tree, one-to many relationship between parent and child records.
- Efficient searching; Less redundant data; Data independence; Database security and integrity
- Disadvantages:
  - Complex implementation
  - Difficult to manage and lack of standards, such as problem to add empty nodes and can't easily handle many-many relationships.
  - Lacks structural independence, such add up application programming and use complexity.
- Handling of spatial Data: No data type available, hence manipulation would happen through front-end as in previous systems. Programming had to be complex, which would thereby have an effect on processing speed, however there is considerable improvement from previous system. Also, less redundant data checks, and better data integrity.

Network data model
- Early 1960s, Charles Bachmann developed first DBMS at Honeywell, Integrated Data Store (IDS).
- It standardized in 1971 by the CODASYL group (Conference on Data Systems Languages).
- Directed acyclic graph with nodes and edges.
- Identified 3 database component: Network schema-database organization; Subschema-view of database per user; Data management language -- at low level and procedural.
- Each record can have multiple parents:
  - Composed of sets relationships, a set represents a one--many relationship between the owner and the member.
  - Each set has owner record and member record.
  - Member may have several owners.
- Main problem: System complexity and difficult to design and maintain; Lack of structural independence.
- Handling of spatial Data: No data type available, hence manipulation would happen through front-end as in previous systems. Programming had to be complex, which would thereby have an effect on processing speed, however this is the first data system which allows multiple users. Though the system becomes more complex, and maintaining the interface is very expensive.

The distinction of storing data in files and databases is that databases are intended to be used by multiple programs and types of users.

Relational database and Database Management System (DBMS)
Based on relational calculus, shared collection of logically related data and a description of this data, designed to meet the information needs of an organization; System catalog/metadata provides description of data to enable program-data independence; logically related data comprises entities, attributes, and relationships of an organization's information. Data abstraction allows view level, a way of presenting data to a group of users and logical level, how data is understood to be when writing queries.

- INGRES at University of California, Berkeley became commercial and followed up POSTGRES which was incorporated into Informix.
- System R at IBM san Jose Lab, later evolved into DB2, which became one of the first DBMS product based on the relational model. (Oracle produced a similar product just prior to DB2.)
- the Entity-relationship(ER) model
- SQL standards
- Object-oriented DBMS (OODBMS)
- Object-orientation in relational DBMSs, new application areas, such as data warehousing and OLAP, web and Internet, Interest in text and multimedia, enterprise resource planning (ERP) and management resource planning (MRP)
- Handling of spatial Data: Even though the earlier models didnot have a standard spatio, or temporal data type but much of the work on the spatial data started from the later versions of the same models. The older versions of various databases adapted to the needs and demands of the current market and are still evolving. Dependency on programming i.e. fron-end is reduced or in some scenarios is neglible for manipulating the spatio, temporal or spatio-temporal data. Specific data types became available in most of the earlier versions of popular RDBMS. Though many models have yet

to incorporate a true spatial interface, but it is a step in the right direction. Complex and real time queries are still cumbersome and relatively slow as compared to queries on non-spatial data.

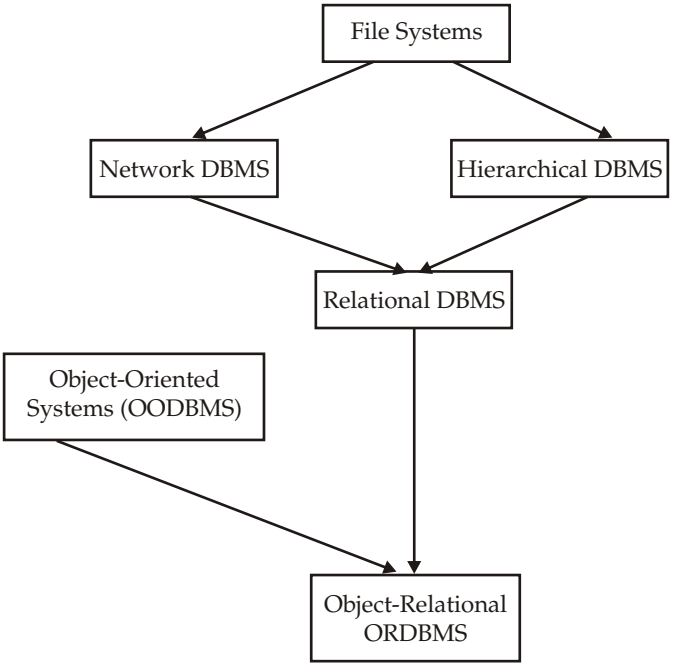


Figure 1: Types of Databases

The Next Phase

Further types of data models comprise of spatial, temporal and spatio-temporal databases. Basically further classification of database is dependent on type of data stored.

- Handling of spatial Data: This work is still in progress. However, true spatial data systems are still evolving and a lot of implementation issues are still being resolved.

Data

But what kind of data is stored? What does data comprise of in a DBMS/RDBMS? Data can be either non-spatial (traditionally), or spatial, temporal, spatio-temporal type:

- Non-spatial data (also called attribute or characteristic data) is that information which is independent of all geometric considerations.
  - For example, a person's height, mass, and age are non-spatial data because they are independent of the person's location.
  - It's interesting to note that, while mass is non-spatial data, weight is spatial data in the sense that something's weight is very much dependent on its location!

- Spatial data includes location, shape, size, and orientation. For example, consider a particular square:
  - its center (the intersection of its diagonals) specifies its location
  - its shape is a square
  - the length of one of its sides specifies its size
  - the angle its diagonals make with, say, the x-axis specifies its orientation.
  - Spatial data includes spatial relationships. For example, the arrangement of ten bowling pins is spatial data.

- It is possible to ignore the distinction between spatial and non-spatial data. However, there are fundamental differences between them:
  - spatial data are generally multi-dimensional and autocorrelated.
  - non-spatial data are generally one-dimensional and independent.

- These distinctions put spatial and non-spatial data into different philosophical camps with far-reaching implications for conceptual, processing, and storage issues.
  - For example, sorting is perhaps the most common and important non-spatial data processing function that is performed.
  - It is not obvious how to even sort locational data such that all points end up nearby, their nearest neighbors.

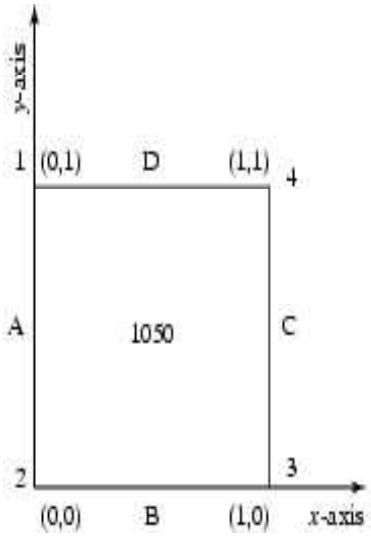
Support of spatial data over the various DBMS

Traditional (non-spatial) database management systems provide:

- Persistence across failures
- Allows concurrent access to data
- Scalability to search queries on very large datasets which do not fit inside main memories of computers
- Efficient for non-spatial queries, but not for spatial queries
- Non-spatial queries:
  - List the names of all bookstore with more than ten thousand titles.
  - List the names of ten customers, in terms of sales, in the year 2001
- Spatial Queries:
  - List the names of all bookstores with ten kms of Ashok Vihar
  - List all customers who live in Delhi and its adjoining states
- Examples of non-spatial data
  - Names, phone numbers, email addresses of people
- Examples of Spatial data
  - Census Data
  - Satellites imagery - terabytes of data per day
  - Weather and Climate Data
  - Rivers, Farms, ecological impact
  - Medical Imaging

Modeling Spatial Data in Traditional DBMS

- A row in the table census_blocks (Figure 2)
- Question: Is Polyline datatype supported in DBMS?



Census_blocks

Name	Area	Population	Boundary
1050	1	1839	Polyline((0,0),(0,1),(1,1),(1,0))

Figure 2: Table census_blocks in a DBMS

Spatial Data Types and Traditional Databases

- Traditional relational DBMS
  - Support simple data types, e.g. number, strings, date
  - Modeling Spatial data types is tedious
- Example: Figure 3 shows modeling of polygon using numbers
  - Three new tables: polygon, edge, points
    Note: Polygon is a polyline where last point and first point are same
  - A simple unit sqaure represented as 16 rows across 3 tables
  - Simple spatial operators, e.g. area(), require joining tables
  - Tedious and computationally inefficient

Mapping "census_table" into a Relational Database

Census_blocks

Name	Area	Population	boundary-ID
340	1	1839	1050

Polygon

boundary-ID	edge-name
1050	A
1050	B
1050	C
1050	D

Edge

edge-name	endpoint
A	1
A	2
B	2
B	3
C	3
C	4
D	4
D	1

Point

endpoint	x-coor	y-coor
1	0	1
2	0	0
3	1	0
4	1	1

Figure 3: Relationships of Table census_blocks in a RDBMS

Spatial Data Types and Post-relational Databases

- Post-relational DBMS
  - Support user defined abstract data types
  - Spatial data types (e.g. polygon) can be added
- Choice of post-relational DBMS
  - Object oriented (OO) DBMS
  - Object relational (OR) DBMS

Spatial Database

A spatial database is a collection of spatial data types, operators, indices, processing strategies, etc. and can work with many post-relational DBMS as well as programming languages like Java, Visual Basic etc.

SDBMS focusses on

- Efficient storage, querying, sharing of large spatial datasets
- Provides simpler set based query operations
- Example operations: search by region, overlay, nearest neighbor, distance, adjacency, perimeter etc.
- Uses spatial indices and query optimization to speedup queries over large spatial datasets.

Components include

- spatial data model, query language, query processing, file organization and indices, query optimization, etc.
- Figure 4 below shows these components

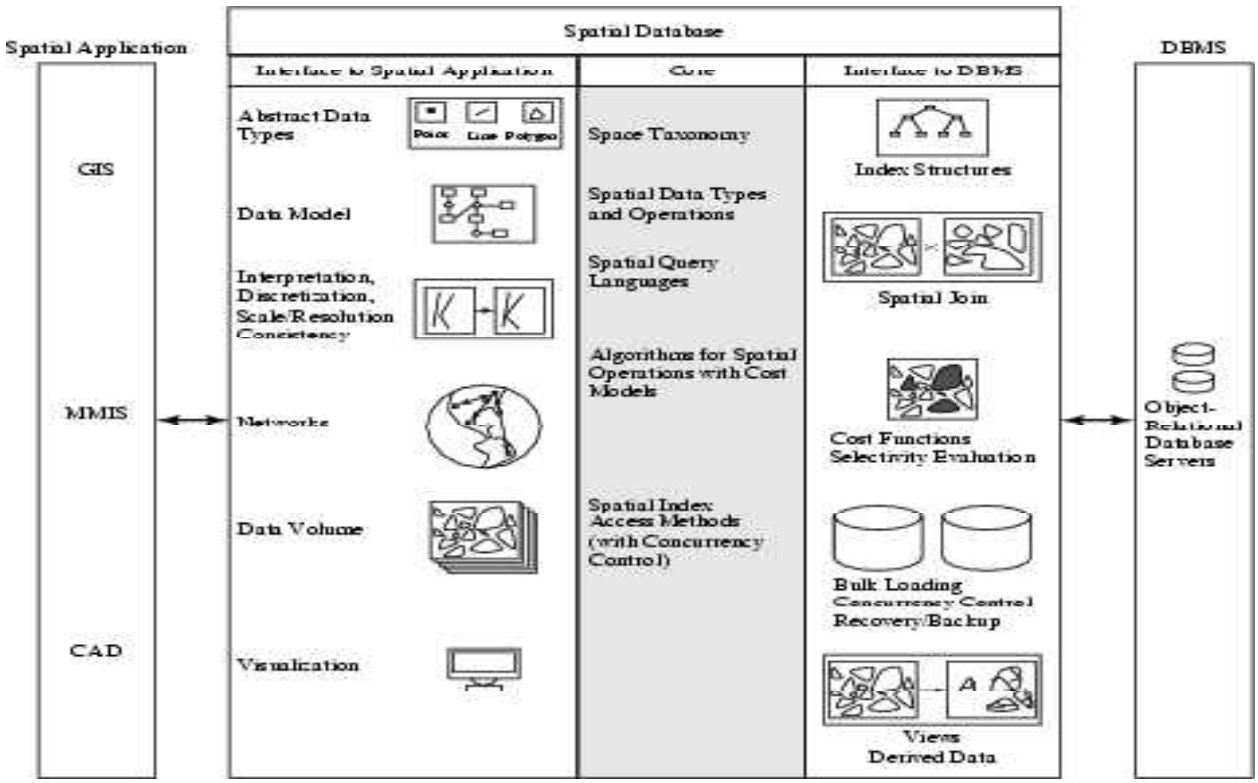


Figure 4: Three Layer Architecture (Source: Adam and Gangopadhyay, 1997)

Implementation of a spatial database

A spatial database works on a three tier architecture as given in figure 4. The Front end is a Spatial application which hides the implementation details and leaves the bulk of work for the backend database. The second tier i.e. the spatial database maps the spatial database with the standard RDBMS elements and acts as an interface between the front-end and the traditional back-end RDBMS. The 3rd tier is a traditional RDBMS or an ORDBMS which doenot have to work on the complexities of the spatial data and treats like any other non-spatial data.

Conclusion

A Spatial database uses an underlying ORDBMS and support for spatial data is provided by all leading RDBMS. Spatial data is sperated from its non-spatial component and is handled seperately by the spatial database which overrides on the ORDBMS. However, as explained above true spatial database are still evolving and are actually overriding a traditional system. As concepts evolve, front-end became easier and back-end took the bulk of task in manipulating data.

References

1. *OpenGIS (1998), "The OGIS specification for extending SQL for geospatial applications".*
2. *Samet (1990), "reference in spatial indexing", Design and analysis of spatial data structure.*
3. *Adam Gangopadhyay (1997), Issues related to the architectural design of spatial databases.*
4. *Chrisman, N. R. (1997), "The Error Component in Spatial Data", www.wiley.com/legacy/wileychi/gis/Volume1/BB1v1_ch12.pdf.*
5. *Robert Laurini (2002), "Real Time Spatio-Temporal Databases", http://onlinelibrary.wiley.com/doi/10.1111/1467-9671.00069/abstract.*
6. *Michael Worboys (2003): "From Objects to EventsModeling the DynamicWorld" http://www.spatial.maine.edu/~worboys/mywebpresentations/events.pdf.*
7. *Egenhofer, Max J., Frank, Andrew U. and Jackson, Jeffrey P.(1989), "A Topological data model for Spatial databases", design and implementation of large spatial, lecture notes in Computer Science, vol 409, pp. 271-286.*
8. *Ramez Elmasri, Shamkant B. Navathe (2003), Fundamentals of Database Systems, 4th Edition.*
9. *Avi Silberschatz, Henry F Korth and Sudarshan, S. (2011), Database System Concepts, 5th and 6th Edition, Mcgraw Hill Publication.*
10. *Raghu Ramakrishnan and Johannes Gehrke (2002), Database Management Systems.*
11. *Stonebraker and Moore 1997, "An overview of the shortcomings of RDBMS in handling spatial data", Object-relational DBMS.*
12. *Guting, (1994), "An overview of spatial databases", http://dna.fernuni-hagen.de/Tutorial-neu.pdf.*
13. *Kriegel (2001), "Integrating a spatial database into off-the-shelf database systems for CAD applications, Kriegel et al.*
14. *De La Beaujardiere (2000), examples of Spatial Database Management System, de La Beaujardiere et al.*