UNIT I DISTRIBUTED DATABASES 9

Distributed Systems – Introduction – Architecture – Distributed Database Concepts – Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing

DISTRIBUTED DATABASE CONCEPTS

WHAT IS DISTRIBUTED DATABASE?

A Distributed database is defined as a logically related collection of data that is shared which is physically distributed over a computer network on different sites. The Distributed DBMS is defined as, the software that allows for the management of the distributed database and make the distributed data available for the users.

A database is an ordered collection of related data. A DBMS is a software package to work upon a database. The three topics covered are database schemas, types of databases and operations on databases.

Database and Database Management System

A database is an ordered collection of related data that is built for a specific purpose. A database may be organized as a collection of multiple tables, where a table represents a real world element or entity. Each table has several different fields that represent the characteristic features of the entity.

For example, a company database may include tables for projects, employees, departments, products and financial records. The fields in the Employee table may be Name, Company_Id, Date_of_Joining, and so forth.

A database management system is a collection of programs that enables creation and maintenance of a database. DBMS is available as a software package that facilitates definition, construction, manipulation and sharing of data in a database. Definition of a database includes description of the structure of a database. Construction of a database involves actual storing of the data in any storage medium. Manipulation refers to the retrieving information from the database, updating the database and generating reports. Sharing of data facilitates data to be accessed by different users or programs.

Examples of DBMS Application Areas

- Automatic Teller Machines
- Train Reservation System
- Employee Management System
- Student Information System

Examples of DBMS Packages

- MySQL
- Oracle

- SQL Server
- dBASE
- FoxPro
- PostgreSQL, etc.

Database Schemas

A database schema is a description of the database which is specified during database design and subject to infrequent alterations. It defines the organization of the data, the relationships among them, and the constraints associated with them.

Databases are often represented through the **three-schema architecture** or **ANSISPARC architecture**. The goal of this architecture is to separate the user application from the physical database. The three levels are −

- **Internal Level having Internal Schema** − It describes the physical structure, details of internal storage and access paths for the database.
- **Conceptual Level having Conceptual Schema** − It describes the structure of the whole database while hiding the details of physical storage of data. This illustrates the entities, attributes with their data types and constraints, user operations and relationships.
- **External or View Level having External Schemas or Views** − It describes the portion of a database relevant to a particular user or a group of users while hiding the rest of the database.

Types of DBMS

There are four types of DBMS.

1. Hierarchical DBMS

In hierarchical DBMS, the relationships among data in the database are established so that one data element exists as a subordinate of another. The data elements have parent-child relationships and are modelled using the “tree” data structure. These are very fast and simple.

2.Network DBMS

Network DBMS in one where the relationships among data in the database are of type many-to-many in the form of a network. The structure is generally complicated due to the existence of numerous many-to-many relationships. Network DBMS is modelled using “graph” data structure.

3.Relational DBMS

In relational databases, the database is represented in the form of relations. Each relation models an entity and is represented as a table of values. In the relation or table, a row is called a tuple and denotes a single record. A column is called a field or an attribute and denotes a characteristic property of the entity. RDBMS is the most popular database management system.

For example − A Student Relation −

4.Object Oriented DBMS

Object-oriented DBMS is derived from the model of the object-oriented programming paradigm. They are helpful in representing both consistent data as stored in databases, as well as transient data, as found in executing programs. They use small, reusable elements called objects. Each object contains a data part and a set of operations which works upon the data. The object and its attributes are accessed through pointers instead of being stored in relational table models.

For example − A simplified Bank Account object-oriented database −

Distributed DBMS

A distributed database is a set of interconnected databases that is distributed over the computer network or internet. A Distributed Database Management System (DDBMS) manages the distributed database and provides mechanisms so as to make the databases transparent to the users. In these systems, data is intentionally distributed among multiple nodes so that all computing resources of the organization can be optimally used.

Operations on DBMS

The four basic operations on a database are Create, Retrieve, Update and Delete.

- CREATE database structure and populate it with data − Creation of a database relation involves specifying the data structures, data types and the constraints of the data to be stored.
- **Example** − SQL command to create a student table −
- CREATE TABLE STUDENT (ROLL INTEGER PRIMARY KEY, NAME VARCHAR2(25), YEAR INTEGER, STREAM VARCHAR2(10));

- INSERT Once the data format is defined, the actual data is stored in accordance with the format in some storage medium.
- **Example** − SQL command to insert a single tuple into the student table −
- INSERT INTO STUDENT (ROLL, NAME, YEAR, STREAM) VALUES (1, 'ANKIT JHA', 1, 'COMPUTER SCIENCE');

- **RETRIEVE** information from the database – Retrieving information generally involves selecting a subset of a table or displaying data from the table after some computations have been done. It is done by querying upon the table.
- **Example −** To retrieve the names of all students of the Computer Science stream, the following SQL query needs to be executed −

- SELECT NAME FROM STUDENT WHERE STREAM = 'COMPUTER SCIENCE';

- **UPDATE** information stored and modified database structure – Updating a table involves changing old values in the existing table's rows with new values. **Example** − SQL command to change stream from Electronics to Electronics and Communications
- UPDATE STUDENT SET STREAM = 'ELECTRONICS AND COMMUNICATIONS' WHERE STREAM = 'ELECTRONICS';

- **Modifying** a database means to change the structure of the table. However, modification of the table is subject to a number of restrictions.
- **Example** − To add a new field or column, say address to the Student table, we use the following SQL command −
- ALTER TABLE STUDENT ADD (ADDRESS VARCHAR2(50));

- **DELETE** information stored or delete a table as a whole – Deletion of specific information involves removal of selected rows from the table that satisfies certain conditions.
- **Example** − To delete all students who are in 4th year currently when they are passing out, we use the SQL command −
- DELETE FROM STUDENT WHERE YEAR = 4;

- Alternatively, the whole table may be removed from the database.
- **Example** − To remove the student table completely, the SQL command used is – DROP TABLE STUDENT;

A distributed database is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.

Features

- Databases in the collection are logically interrelated with each other. Often they represent a single logical database.
- Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites.
- The processors in the sites are connected via a network. They do not have any multiprocessor configuration.
- A distributed database is not a loosely connected file system.
- A distributed database incorporates transaction processing, but it is not synonymous with a transaction processing system.

Distributed Database Management System

A distributed database management system (DDBMS) is a centralized software system that manages a distributed database in a manner as if it were all stored in a single location.

Features

- It is used to create, retrieve, update and delete distributed databases.
- It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users.
- It ensures that the data modified at any site is universally updated.
- It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously.
- It is designed for heterogeneous database platforms.
- It maintains confidentiality and data integrity of the databases.

Factors Encouraging DDBMS

The following factors encourage moving over to DDBMS −

- **Distributed Nature of Organizational Units** − Most organizations in the current times are subdivided into multiple units that are physically distributed over the globe. Each unit requires its own set of local data. Thus, the overall database of the organization becomes distributed.
- **Need for Sharing of Data** − The multiple organizational units often need to communicate with each other and share their data and resources. This demands common databases or replicated databases that should be used in a synchronized manner.
- **Support for Both OLTP and OLAP** − Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) work upon diversified systems which may have common data. Distributed database systems aid both these processing by providing synchronized data.
- **Database Recovery** − One of the common techniques used in DDBMS is replication of data across different sites. Replication of data automatically helps in data recovery if the database in any site is damaged. Users can access data from other sites while the damaged site is being reconstructed. Thus, database failure may become almost inconspicuous to users.
- **Support for Multiple Application Software** − Most organizations use a variety of application software each with its specific database support. DDBMS provides a uniform functionality for using the same data among different platforms.

Advantages of Distributed Databases

Following are the advantages of distributed databases over centralized databases.

- **Modular Development** − If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning. However, in distributed databases, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions.
- **More Reliable** − In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.
- **Better Response** − If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. On the other hand, in centralized systems, all queries have to pass through the central computer for processing, which increases the response time.
- **Lower Communication Cost** − In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized. This is not feasible in centralized systems.

Adversities of Distributed Databases

Following are some of the adversities associated with distributed databases.

- **Need for complex and expensive software** − DDBMS demands complex and often expensive software to provide data transparency and co-ordination across the several sites.
- **Processing overhead** − Even simple operations may require a large number of communications and additional calculations to provide uniformity in data across the sites.
- **Data integrity** − The need for updating data in multiple sites pose problems of data integrity.
- **Overheads for improper data distribution** − Responsiveness of queries is largely dependent upon proper data distribution. Improper data distribution often leads to very slow response to user requests.

Types of Distributed Databases

Distributed databases can be broadly classified into homogeneous and heterogeneous distributed database environments, each with further subdivisions, as shown in the following illustration.

Homogeneous Distributed Databases

In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are −

- The sites use very similar software.

- The sites use identical DBMS or DBMS from the same vendor.
- Each site is aware of all other sites and cooperates with other sites to process user requests.
- The database is accessed through a single interface as if it is a single database. Types of Homogeneous Distributed Database

There are two types of homogeneous distributed database −

- **Autonomous −** Each database is independent that functions on its own. They are integrated by a controlling application and use message passing to share data updates.
- **Non-autonomous −** Data is distributed across the homogeneous nodes and a central or master DBMS coordinates data updates across the sites.

Heterogeneous Distributed Databases

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models. Its properties are −

- Different sites use dissimilar schemas and software.
- The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.
- Query processing is complex due to dissimilar schemas.
- Transaction processing is complex due to dissimilar software.
- A site may not be aware of other sites and so there is limited co-operation in processing user requests.

Types of Heterogeneous Distributed Databases

- Federated − The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.
- Un-federated − The database systems employ a central coordinating module through which the databases are accessed.