Master of Computer Applications (MCA)

Database Management System (DMCACO102T24)

Self-Learning Material (SEM 1)

Jaipur National University

Centre for Distance and Online Education

Established by Government of Rajasthan

Approved by UGC under Sec 2(f) of UGC ACT 1956

&

NAAC A+ Accredited

PREFACE

In this guide, we will explore the fundamental concepts, principles, and best practices for effectively managing databases. Whether you are new to database management or looking to enhance your existing skills, this resource will provide valuable insights and guidance to help you successfully navigate the complex world of data storage and organization.

Databases are essential components of many applications and systems, and their proper design and management are crucial for ensuring efficiency, data integrity, and scalability. Through this preface, we invite you to embark on a journey of discovery and exploration into the world of databases.

By understanding the importance of database management, you will be equipped to make informed decisions, optimize database performance, ensure data integrity, and promote efficient data retrieval and analysis. Through this preface, we invite you to embark on a journey of discovery and mastery of database management principles that will empower you to harness the full potential of your data assets.

TABLE OF CONTENTS

UNIT 1

BASICS OF DATABASE

Learning Outcomes:

- Students will be capable of learning about the database and database users.
- The features of databases and database systems will be understood by students.
- It will be possible for students to comprehend the ideas and design of DBMS.
- Pupils will be capable of comprehending the concepts of instances and schemas.
- The topics of database languages and interfaces will be taught to the students.

Structure

1.1 Database and Database User

The majority of us deal with databases on a regular basis, making databases and database systems an essential component of modern life. A database is most likely accessed by someone or some computer programme when we use a bank to deposit or withdraw money, make reservations for lodging or travel, search for a bibliographic item in an automated library catalogue, or make an online purchase (like a computer, toy, or book). The database that houses the inventory of grocery goods is immediately updated at the supermarket whenever you make a purchase. People's computer usage is greatly influenced by databases and database technologies.

A collection of software applications called database management systems (DBMS) let users build and maintain databases. General-purpose software programs known as database management systems (DBMS) assist users and applications in defining, creating, modifying, and sharing databases. Defining a database involves outlining the types of data, formats, and limitations that will be used for storing information in the database. The DBMS also keeps database definitions and descriptive data stored as meta-data in dictionaries or database catalogues.

Building a database involves storing data on a storage device managed by the database management system.

Databases play a crucial role in almost every aspect of computer-related fields such as business, electronic commerce, engineering, medical, genetics, law, education, and librarianship. This is because the term "database" is used commonly. We must first explain it. A database consists of interconnected data. Data is information that consists of verifiable facts with inherent meaning that can be recorded. Take into account the names, phone numbers, and addresses of your contacts. This data could have been saved on a hard drive or in a searchable address book on a personal computer, utilizing software such as Microsoft Access or Excel. A database is a structured collection of information or data that is typically stored electronically in a computer system. Databases are usually divided into two main types: relational or sequential databases, and non-relational databases.

- A database is an organized set of connected data with a primary goal. The following are a few implicit properties of the database:
- A database is a representation of a subset of reality, sometimes called the spoken universe or the mini-world. The database will update in response to changes made to the mini-world. A database is a collection of logically consistent data that has some underlying meaning.
- A database has a target audience and specified applications in which the audience is interested. It is designed, constructed, and loaded with data for a specific purpose.

To put it another way, a database has an audience that actively interacts with its contents, a data source from which it obtains information, and some degree of contact with actual occurrences. The information in the database may change as a result of events or business actions that end users partake in, such as a customer buying a camera or an employee having a baby.

To be accurate and reliable, a database needs to be a true reflection of the mini-world it represents at all times; hence, changes need to be reflected in the database as soon as they happen.One of the most well-known instances of a large corporate database is Amazon.com. More than 20 million books, CDs, movies, DVDs, games, gadgets, clothes, and other products are covered in its database. The database is typically spread among 200 machines, or "servers," and has a size of over 2 terabytes (1012 bytes). Approximately 15 million individuals visit Amazon.com daily to make purchases from its inventory. When new books and other products are added to the inventory, the database is updated often, and stock counts are updated as transactions occur. The Amazon database will be updated with the assistance of over 100 people. There are two ways to create and manage a database: manually and automatically. An example of a manual database is a library card catalog. Create and manage a computerized database with a database management system or a set of application programs made specifically for that purpose.

Database Users

Individual objects that describe a set of fields and business rules can be read, inserted, edited, and deleted by database users. One or more database tables may also be updated using these objects. Choose the Database Access action in the Users app to create database users. Database users in DBMS can utilize the applications of the Database Management System to access and retrieve data from the database. We can always keep our database safe and stop unauthorized users from accessing it. distinct database users with distinct login IDs and passwords are only able to access the database areas that they are normally associated to, based on certain criteria. DBMS database users can be categorized based on how they communicate with the database.A role is a type of database item that has the ability to grant users one or more rights. A user is granted all of the privileges associated with the role they are assigned. A user can play multiple roles. There's also a role hierarchy accessible.

Could you provide a database user example?

For example, users who purchase train tickets are not very experienced. Even though they don't comprehend DBMS, bank clerks use the database to do their given responsibilities, making them natural users. A user who assesses parametric end users' demands is called a system analyst.

Based on the tasks they perform on the databases, database users can be divided into seven types, which are as follows:

- **Database Administrators (DBA)**

The Database Administrator (DBA) is the most important type of database user in DBMSs. A database administrator is an individual or group of individuals who creates the database schema and oversees the database at various organizational levels. Super-users, often known as database administrators (DBAs), are in complete control of the database. They collaborate with programmers to plan and create the database's overall structure, functionality, and process. DBAs have the authority to issue or remove authorization permissions for any other user at any moment.

- **Database Designers**

Database designers, as the name suggests, are users of database management systems (DBMSs) who plan and construct the database's structure, which includes tables, constraints, entity relationships, triggers, indexes, and schemas. Before creating the final architecture of the database so that programmers can write its logic, database designers try to collect information based on database needs such as layout, appearance, functionality, costing, technologies to be employed, and implementation approaches. The DBMS database users who are in charge of creating the entire database design are known as database designers. They decide what type of attributes will be used, what kind of data has to be saved, what sort of links exist between database entities, and so on.

- **System Analysts**

Using DBMS databases, system analysts examine the needs of naive and parametric end users. They have to determine if all end-user requirements have been met. Among the primary responsibilities of a system analyst in DBMSs are feasibility, economic, and technical analysis.On sometimes, they are in charge of the use, arrangement, and design of the database. Usually, they check and collect all pertinent database information, and if needed, they can alter or update the database's final design to meet the requirements. System analysts make ensuring that all standards are met by the final output.

• A systems analyst uses their analytical and design abilities to use information technology to address business problems. Systems analysts can take on the role of change agents by determining what organizational improvements are required, creating the systems to carry out those changes, and inspiring others to adopt the systems.

- **Application Programmers / Back-End Developers**

Application programmers, sometimes referred to as back-end developers, are specialists in computers who write user interfaces and application programs (in C, C++, Java, PHP, Python, and other languages) so that other users can use these apps to communicate with the database.
Database programmers are familiar with database management systems. DML (Data Manipulation Language) queries that interface with the database are used to store and retrieve data. If necessary, application programmers can explain modifications to the database architecture of an application. They are able to create databases in any language they are familiar with.

- **Naive Users / Parametric Users**
  - Naive users, sometimes referred to as parametric end users, are those who frequently use database applications to accomplish their goals but have little to no familiarity with

database management systems. Using the interface provided by DBMS programs, novice users frequently use the database to enter or retrieve data. Because they can interact with the database system through a menu-driven application interface, inexperienced users do not need to be aware that it exists. Naive / Parametric End Users connect directly to the database via the created applications in order to get the needed results. Users who book tickets for trains, for instance, are naive/parametric end users since they usually don't know how to utilize database management systems (DBMS) and rely mostly on the train booking application.

- **Sophisticated Users**
  - Sophisticated database users are familiar with the database and comprehend DBMS (DDL and DML commands). System analysts, engineers, scientists, and business analy sts need to be considered sophisticated users. • Database applications can be created and accessed by knowledgeable individuals without the need to write program code. This is due to the fact that these users—also known as SQL programmers—can communicate directly with the database through SQL queries and query processors. Information can also be taken out of a database using SQL queries. Additionally, they can be used to update, add, or remove data from the database.

Casual Users / Temporary Users

Database customers who rarely use the database services are referred to as transient or casual users. These users want all of the information to be available in one location when they try to access the database. Because they are unfamiliar with DBMS, casual or transient users need to enter new data each time they want to access the database. For instance, high-level management staff members are sporadic users with little knowledge of DBMSs who can use the database to add new information or retrieve previously collected data.

1.2 Characteristics of the Database

- **Self-describing nature of a database system**
  - A self-describing database system needs metadata that specifies and explains the data and relationships between tables in addition to the database itself. This information can be

used by database or DBMS software users as needed. A database system differs from a traditional file-based system in that it separates data and data about data.

- **Insulation between program and data**
    - Because application programs establish the format of the data files in a file-based system, changing the structure of a file may require changing all programs that access it. In contrast, the database approach keeps the data structure in the system catalogs instead of the programs. Because of this, changing a file's structure simply requires one modification. The term "program-data independence" typically refers to the separation of data and programs.
- **Support for multiple views of data**
    - Multiple data perspectives are offered by databases. A view is a preconfigured portion of the database that particular system users can only see. It is possible for various persons to view the system from different angles. Only information pertinent to one or more users may be included in each view.
- **Enforcement of integrity constraints**
    - For accurate data entry and data integrity, database management systems need to let users set and enforce specific limitations. A database constraint is a restric tion or guideline that controls what can be added to or changed in a table. Examples of constraints include the use of a particular postal code type or the entry of an authorized city in the City field. Constraints on databases can take many different forms. The data type, for instance, establishes what kind of information—such as just numbers—can be entered into a field. To ensure that no data is entered twice, use the primary key. Limitations might be straightforward (field-based) or complex (programming).
- **Sharing of data and multiuser system**

Databases are now designed to allow several people to access them. They therefore allow multiple users to view the same database simultaneously. This access is possible due to concurrency control mechanisms. These strategies ensure that data integrity and correctness are consistently maintained. Today's multiuser database systems outperform previous single-user databases.

- **Data redundancy Control**
    - The database should preferably only save each data product once. However, it is also regulated by the application programming and is utilized to keep the database as simple

as possible by minimizing redundancy. Data redundancy is occasionally employed to improve system performance.

- **Data Sharing**
  - There are many advantages to integrating all of an organization's data into a database system. First, it permits employees and other system users to share data. Secondly, it facilitates the extraction of more information than would be feasible from a given volume of data by users.
- **Restriction of unauthorized access**
  - Another benefit of a database management system is data independence. Put differently, system data descriptions or data describing data (metadata) are not the same as application programs. This is also plausible since database management systems, as opposed to being integrated into the application itself, might be able to handle modifications to data structures.
- **Data independence**
  - Another advantage of a database management system is data independence, or the notion that application programs are independent of system data descriptions or data describing data (metadata). This is made possible by the fact that database management systems, as opposed to being integrated into the application itself, handle modifications to data structures.
- **Transaction processing**
  - Concurrency control subsystems are typically needed for database management systems. This feature guarantees that, even when numerous users are updating the same piece of information at the same time, the data is typically accurate and consistent during transaction processing.
- **Provision for multiple views of data**
  - Concurrency control subsystems are necessary for a database management system. This feature makes sure that, even when numerous users are altering the same piece of information at the same time, the data is typically accurate and consistent during transaction processing.
- **Backup and recovery facilities**

Techniques for preventing data loss include recovery and backup. The database system provides not only a network backup but also an independent procedure for data recovery and backup. Restoring a database from a backup is the sole method available for recovering data that was lost due to a failing hard drive.

1.3 Database Systems

The software components known as database systems, or database management systems (DBMS), gather digital and electronic records, extract relevant data from them, and store it.A typical database's objective is to store and retrieve data. For example, structured data processing and storing are the primary functions of standard relational databases. SQL, or structured query language, is generally used by databases to access the data that is kept in their tables.

Processing big, quick, and varied datasets is facilitated in part by databases and database systems. Businesses won't be able to obtain insightful data and in-depth analytics without the Database Management System. Business organizations may carry out a variety of data-processing tasks thanks to the database environment, which enables data to be accessed, changed, regulated, and then presented in an orderly manner.

Well liked DBMSs include Oracle Database, FileMaker Pro, Microsoft SQL Server, MySQL, and dBase. Usually, data is arranged in rows and columns to ease the burden on the worker and provide fast, precise results. Data of many kinds, including binary, textual, time series, and numerical data, can be processed, stored, and retrieved by a database management system.

1.4 Concepts and Architecture

1.4.1 Concepts of DBMS

A database is an organized collection of facts or information that is typically kept electronically in a computer system. A database management system often oversees a database. Tables, keys, and relationships are the three main components of a relational database. Database architecture builds specialized applications for businesses or organizations using computer languages. Database architecture is the design, development, implementation, and maintenance of computer programs that store and organize data for businesses, organizations, and institutions. A database architect designs and develops software to meet customer requirements. A DBMS's architecture influences its design. It could be decentralized, centralized, or hierarchical in structure. A DBMS's architecture can be divided into single-tier and multi-tier categories.

Tiers are Categorized as follows:

- **Architecture with a single tier**

  In a one-tier or single-tier architecture, all of the required components for a software application or technology are placed on a single server or platform. In essence, a one-tier architecture will keep all of an application's elements, such as the interface, Middleware, and back-end data, in a single place. These systems are usually regarded by the developers as one of the simplest and most direct method.

- **Architecture with two levels**

  Client Server is a two-tier architecture. A client-server application is similar to a two-tier architecture. The client and the server have direct communication. Between the client and the server, there is usually no intermediary.

- **Three-tiered architecture**
  - The intricacy of the users and the ways in which they can utilize the database's contents dictate the levels of a three-tier architecture. It is the most widely used architecture for building a DBMS. There are several uses for this architecture. It can be applied to distributed applications as well as the. Distributed systems benefit greatly from the application of this architecture.

- **Architecture with n levels**
  - The hierarchy of a three-tier system is determined by the complexity of the users and how they use the data in the database. It is the design that is used in DBMS building the most frequently. This architecture has a wide range of applications. Web and distributed apps can both use it. This architecture works especially well in distributed systems.

1.4.2 Database Architecture

A database management system's (DBMS) design shows how users interact with data within a database. It doesn't care how data is handled or processed by the database management system. It supports the development, construction, and upkeep of databases used by corporations to store and arrange information. The architecture of a database management system typically dictates its fundamental idea. It is feasible to create hierarchical, decentralized, and centralized structures

The three levels of DBMS architecture are as follows:

- **External tiers.**
  - The external level is a portion of the database that pertains to each particular user. At this level, conceptual and internal level details are shielded from users. The greatest degree of database abstraction is this one. It has many user views and external schemas. A single user or a group of users can access many views of the same database using external layers. An external view offers a strong and adaptable security solution by preventing the specific user from accessing some of the most fundamental database components.

- **Conceptual tiers.**
  - • The kinds of data that are normally stored in the database and their relationships are frequently described at the conceptual level.

It stands for the following:

- ✓ Information about all the entities, attributes, and relationships.
- ✓ The limitations of the data.
- ✓ Information about security and integrity.

- **Internal tiers.**
  - The physical representation of the database on the computer is known as the internal level. This level explains the standard procedure for storing data in a database. It covers file organization and data structures that are used to store different types of data on storage systems. The graphic below shows the different DBMS architecture tiers.

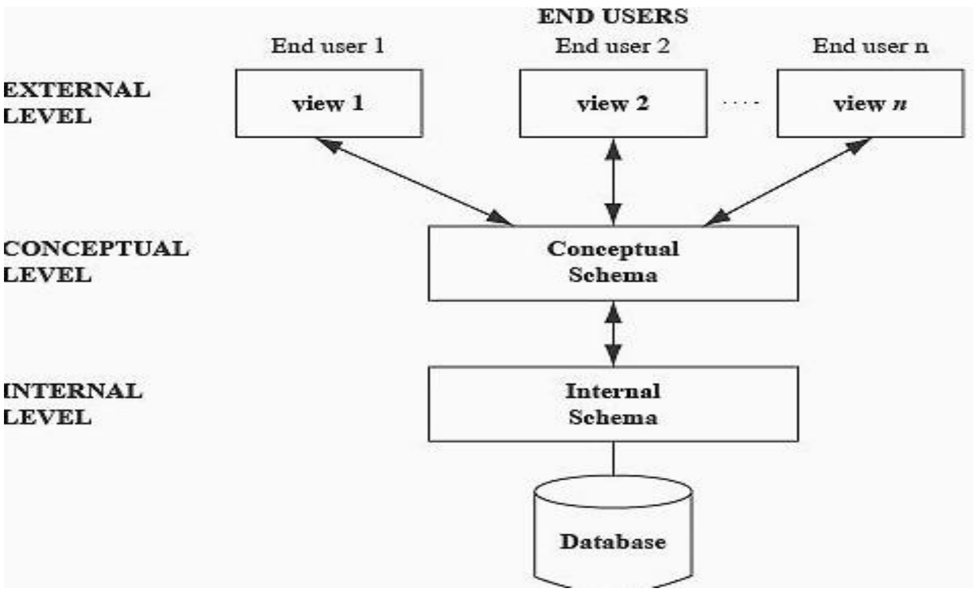


Fig 1.1 **DBMS architecture**

- **Knowledge Check 1**

  Fill in the Blanks

1. The architecture of the Database Management System (DBMS) shows how _______ interacts with the data in the database.

2. ________ database users in the DBMS who will examine the needs of the end users, either parametric or naive.

- **Outcome-Based Activities 1**

  Try to Illustrate the architecture of Database Management System.

1.5 Schema and Instances

Though there is a big distinction between Schema and Instance in DBMS, both of them help describe the data that is available in a database.Any given database's general description is referred to as its schema. The collections of data and information that the database keeps on file at any one time are referred to as instances. The general description of the database is known as the schema. The fundamental framework for data storage in a database is known as a schema. The general layout of the database is referred to as the database schema. There won't be many updates to the schema. It is described as the logical organization of the database. The database data is not displayed.

The following are the different kinds of schemas:

The term "physical schema" describes a database's actual physical layout. It is described as easily changeable and concealed beneath the logical schema so as not to interfere with any application programs.

Schema logical The database design is referred to as a logical schema. Using the logical structure, programmers create the apps.

The schema at the view level is known as the external schema. In a schema, the views intended for end users are often defined at the top level.

Typically, the Database Management System (DBMS) allows for the provision of several sub or external schemas in addition to one logical and physical schema. The database schema is the structure and format of the database where the data is typically kept. Until something else is changed, it is the one thing that stays the same. It often outlines the data's structure and intended method of storage.

Name, email, phone, and address fields are what a person's database schema will include, as you can see below.

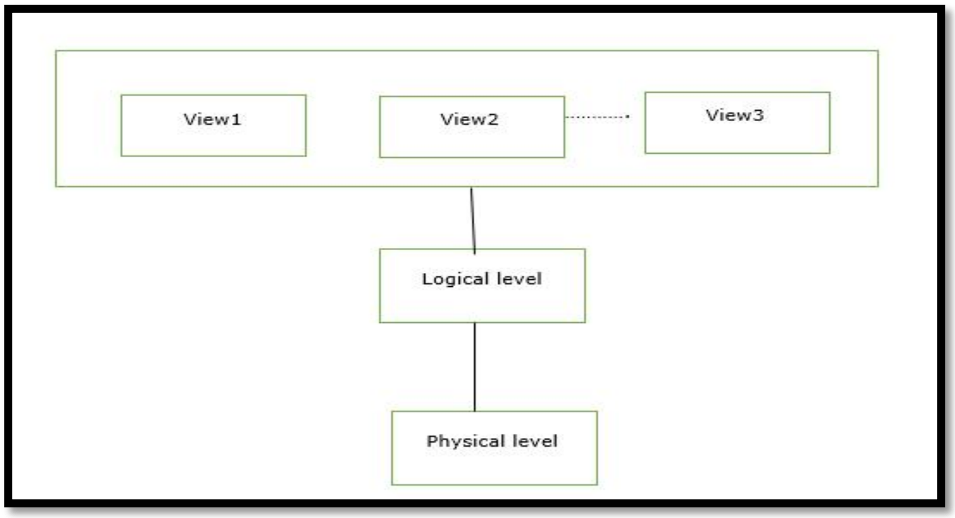


Fig. 1.2 Format of database

Instances

These are a compilation of all the data and information that is kept on hand at any given time. CRUD operations, such as data and information deletion and addition, can be used to quickly change these instances. It's crucial to remember that no search results result in any changes.

The instances can be changed using specific CRUD operations, like data addition and deletion. It should be noted that no search will alter the instances in any way.

Assume that we have 50 records and a teacher in a table named "School" in our database. This indicates that the database instance presently has 50 records, and we plan to add an additional 50 records tomorrow, for a total of 100 records. This case is the one we are discussing.

This is depicted below:

Table 1.1 Storing data in data base

Name	Email	Phone no
BOB	Kysd@yasd.com	2343489
JAY	wwr@sdas.in	5345476
PRIYANKA	wer@sdf.com	2342349

1.6 Data Independence

The ability to make changes to the scheme without affecting previously written progress and applications is known as data independence. Programs and data are maintained apart to ensure that modifications to one won't impact the functionality of the other. Data independence is the main objective of all three layers of data abstraction, as we already know. Changes made at one level must not affect data at other levels if the database expands and changes over time. Updating the database would therefore require less resources and time.

There are three levels of abstraction and two levels of data independence.

These are the following:

- **Physical data Independence**

  When we talk about physical data independence, we mean the capacity to modify the physical level without modifying the logical or conceptual levels. We can modify the database's storage system using this property without having an impact on the logical schema.

  The following given techniques may be used to effect changes at the physical level.

• A brand-new magnetic tape or hard disk for storage.
• A brand-new storage data structure. utilizing a different method of data access or file organization
• Changing the database's location.

The impartiality of rational information

Logical Data Independence

The user's perspective on the data is the logical view of it. It gives users information in an understandable fashion. Users ought to be able to change the Logical View of data without being aware of the physical location of the data, in accordance with Codd's Rule of Logical Data Independence. A computer application or piece of software modifies the data's logical view. Decisions regarding what data should be kept in the database and how to apply the logical degree of abstraction are made by the database administrator.. It provides a comprehensive grasp of Data. It also explains the relationship and the information that will be stored in the database.at a base has a simple structure because of the data in dependence. It relies on application domain entities to fulfill functional requirements. It abstracts the functional requirements of the system. The class object diagrams define the static structure for the logical view. Users are not permitted to alter the database's logical structure.

Rationale modifications might comprise.
• Adjust the definition of the data.

• Changing, removing, or adding any new entity, relationship, or attribute to the database.

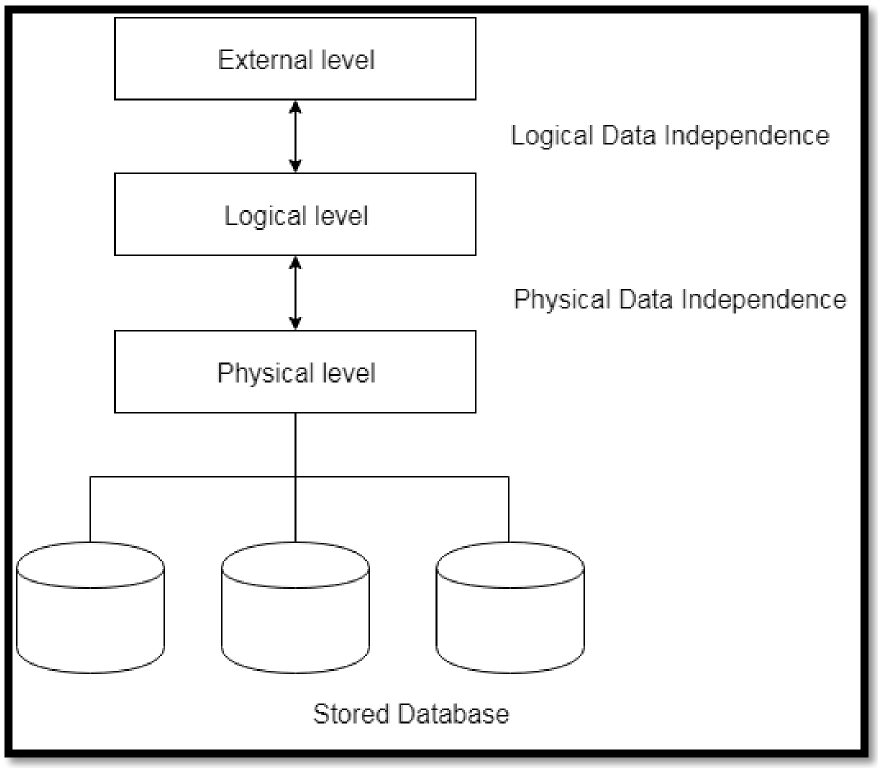


Fig 1.3 Data manipulation in data base

Database Languages

The appropriate languages and interfaces for expressing database modifications and queries are provided by DBMS. Data in a database can be read, stored, and updated using database languages. Data must be altered by insertion, deletion, updating, and modification after it has been stored or filled. For these operations, the database management system offers a number of languages (DBMS). Thus, reading, updating, and storing data in a database are all done using database languages.

- Create, Drop, Truncate, and Rename are some of the Data Definition Language (DDL) commands.
- Data Manipulation Language commands are Select, Insert, Delete, and Update.
- Revoke, Grant are DCL commands.
- TCL commands are Rollback, Commit are the TCL commands.

1.7 Interfaces

The functionality of a DBMS is abstracted in a DBMS interface. It typically refers to the line of communication between a DBMS and its clients or to the abstraction that a DBMS component offers. The implementation of the encapsulated component's functionality is hidden behind a DBMS interface. Using a database management system (DBMS) interface, you can enter queries into a database without knowing the query language.

The following user interfaces may be offered by DBMSs:

Menu-Based Web Client or Browsing Interfaces

These user interfaces provide the user with an options menu to assist them in creating a request. Using menus relieves the burden of having to memorize specific instructions and query language syntax, which is its main benefit.

By compiling or choosing options from a menu that the system displays, the query is built step-by-step. In Web-based interfaces, pull-down menus are a common design element.

Forms-Based Interfaces

Every user is presented with a form via a forms-based interface. To add new data, users can choose to fill out the complete form or only a piece of it. In the latter instance, the DBMS will use the same type of data for the remaining fields. For users who are not familiar with operating systems, these kinds of forms are typically created, designed, and programmed. Forms specification languages are special languages that assist in the specification of such forms, and they are present in many DBMS. For example, a form-based language such as SQL* Forms uses a form designed in tandem with the relational database schema to specify queries.

Graphical User Interface

A schema is typically presented to the user in diagrammatic form via a graphical user interface (GUI).After that, the user can alter the image to include a question. Forms and menus are frequently used in GUIs.A pointing device, like a mouse, is typically used by graphical user interfaces (GUIs) to choose a particular area of the displayed schema diagram.

Natural language Interfaces

Natural language interfaces recognize and try to understand requests that are sent in a language other than English. A dictionary of keywords is also included in the conceptual schema of a natural language interface, which is similar to that of a database.

Speech Input and Output

Speech is being used less frequently, but it is becoming more widespread and general—whether it is in response to a request, a question, or an answer to one.

In applications with restricted vocabulary, such phone directory inquiries, airplane arrival/departure information, and bank account information, speech is typically permitted for input and output purposes, making this information accessible to regular people.

DBA interfaces –

Most of the database systems have exclusive DBA staff-only privileged commands. These instructions include those for establishing system parameters, authorizing accounts, modifying schemas, and rearranging all the database storage structures.

- **Knowledge Check 2**

  State true and false for the following sentences

  - 1. Database clients are used to access, store, and update data in databases. After being filled out or stored, data has to be updated, modified, deleted, and manipulated.
    2. A program interface known as a database management system (DBMS) interface enables you to submit queries into a database without needing to understand the query language.
    3. A schema is usually shown to the user in diagrammatic form through a graphical user interface (GUI).

- **Outcome-Based Activities 2**

  - Use an appropriate illustration to demonstrate the distinction between logical and physical data independence.

1.8 Summary

- Databases are organized collections of data that are typically kept electronically within computer systems. Typically, a database management system is used to administer a database (DBMS).
- Within DBMS, a database user is any individual who utilizes and has access to a database. Data access and retrieval from databases are made possible by applications and interfaces provided by the Database Management System (DBMS).

- Database users are often authorized to read, write, edit, and remove particular objects that specify a collection of fields and business rules. One or more database tables can also be updated by these items. Database users are created using the Database Access action in the Users application.
- The general description of a database is referred to as its schema. A database stores a collection of data and information at any one moment, which is essentially what an instance is. The entire database is still using the same schema.
- One kind of programming language used to define and work with databases is database programming language.The four types are DDL, DML, DCL, and TCL.
- The metadata and schema of the database can be created or altered using DDL commands.Data that has previously been stored in schema objects can be accessed and modified using DML commands.
- A query language is not necessary when using a database management system (DBMS) interface, which is a user interface that lets you enter queries into a database.

1.9 Self-Assessment Questions

1. Describe a database
2. What categories of database users exist?
3. What features does the database have?
4. What is Database Systems?
5. Explain the concept and architecture of the Database.
6. What is schemas and instances?
7. What is Data Independence?
8. What are the different types of DBMS Languages?

1.10 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: PergamumInfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.

- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 2

RELATIONAL DATABASE DESIGN

Learning Outcomes:

- Students will be capable of learning about Functional Dependencies and its implication.
- Knowledge of normalization and closure rules will be attained by the students.
- The ideas of the Decomposition and Synthesis approach will be understood by the students.
- The 3NF and BCNF will be understood by the students.
- The Lossless join and Dependency Preserving decomposition will be taught to the students.
- Pupils will gain knowledge about 4NF and Multi-valued dependence.
- Students will understand the concept of Join dependency and 5NF.

Structure

2.1 Functional Dependencies and its implication

FD is a constraint in DBMS that defines the link in between an attribute and another attribute. The data integrity of the database is maintained with the help of functionaldependency. Understanding the distinction between poor and excellent database design is essential. A functional reliance is shown as a "arrow". Y represents how functionally dependent X is on Y.

X -> Y

It is known what the functional dependency's left side is. Assume that the columns Emp Id, Emp Name, and Emp Address are present in our employee table.

This is due to the fact that, if we know the employee name associated with an Emp Id, we may ascertain it; in this instance, the Emp Id property can be utilized to specifically identify the Emp Name attribute of the employee table. The definition of functional dependability is:

Emp Id->Emp Name

In the above example,Emp_Name is functionally dependent on Emp Id.

2.1.1Types of Functional Dependency

1. Trivial functional dependency

The functional dependence “A->B” is straight forward if “B” is a subset of “A”. Trivial dependencies include, for example, “A -> A” and “B ->B.”

2. Non-trivial functional dependency

If B is not a subset of A, then there is a non-trivial functional dependency between A and B.We say that A and B are completely non-trivial when their intersection is NULL..

1. ID $\rightarrow$ Name,
2. Name $\rightarrow$ DOB

2.2 Closure rules

The collection of all the characteristics that are typically functional dependencies on attribute X with respect to the F is known as the closure of an attribute X. Typically, it is indicated by the X+, which stands for what X can typically discern. The closure of an attribute is a collection of additional attributes whose functionality may be deduced from it.

When one can infer the set “F+” of all FDs from “F”, a group of FDs is said to be closed. Closure is the set “X+” of all attributes that, with respect to F, are functionally determined by a set of attributes X.

Algorithm

Let's examine the X+ computation algorithm.

First 1: X+ = X

Step 2: iterate until X+ stays the same.

For each FD in F, Y->Z

If Y ⊆ X+ then X+ = X+ U Z

Pseudocode

The closure of X under functional dependency set F is X+, so ascertain this.

X Closure: = X itself will be contained;

Continue the procedure as follows:

X closure = old X Closure;

Do this for each functionally related P and Q in the FD set.

X Closure: = X Closure U Q if X Closure is a subset of P.

Until (X closure = old X Closure), repeat;

2.3 Normalization

Normalization is the process of arranging data in a database. This includes creating tables and establishing linkages between them in line with rules meant to protect the data and improve database flexibility by getting rid of inconsistent dependencies and superfluous information.

Redundancy in a relationship or set of relationships is decreased by normalization. Update, insertion, and deletion anomalies may result from relational redundancy. It reduces relational redundancy as a result. Redundancy in database tables is reduced or eliminated by using standard formats. "Normal forms" refer to stages of normalization. The traditional forms deal with interpersonal connections. If a relationship satisfies certain requirements, it is said to be in a certain normal form.

Normalization is mostly used to get rid of data abnormalities. Data redundancy is the main reason anomaly elimination fails, and as databases get bigger, it makes other problems like data integrity worse.

The process of organizing the data in a database is called normalization. The goal of normalization is to reduce redundant information in a relationship or set of relationships. Eliminating undesired features like Insertion, Update, and Deletion Anomalies is another application for it. The bigger table is split up into smaller ones during normalisation, and these are connected via relationships.

When there is insufficient information to permit the addition of a new tuple to a relationship, this is known as an insertion anomaly

. When important data is inadvertently destroyed along with other data, this is referred to as a "deletion anomaly".

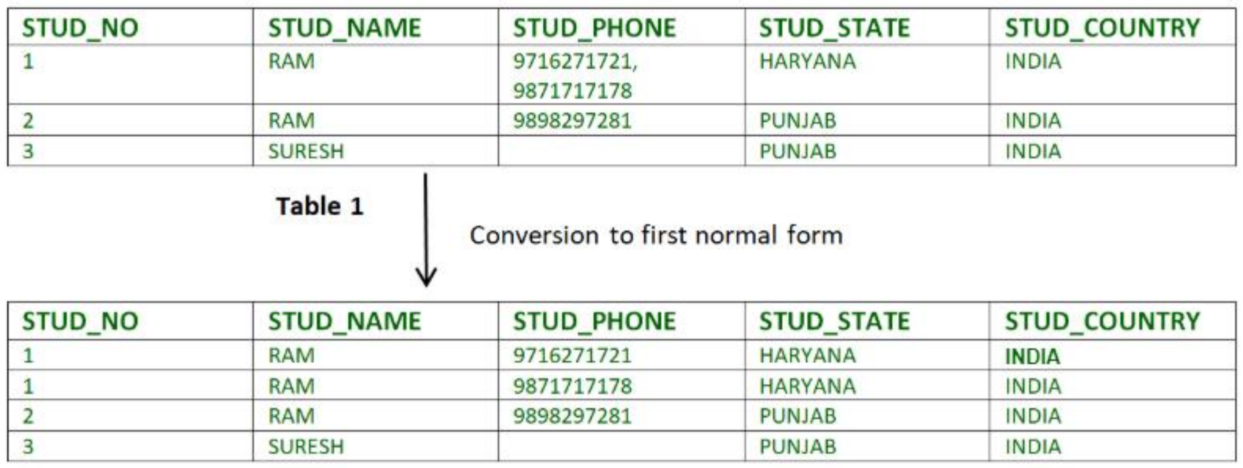
An update anomaly occurs when changing a single data value necessitates changing many data rows.

2.3.1 Types of Normal Form

1.First Normal Form

When a relation has an single value, it is referred to in DBMS as being in 1 normal form, or 1NF. To put it another way, 1NF means that a table attribute can only contain single value and not more than one.
A relation is in the first normal form if it lacks any composite or multi-valued key qualities; if it does, the first normal form is broken.
A connection is said to be in first original form if every attribute in the relation is a single-valued attribute.

Table- 2.1 Data storage



STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	INDIA
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

2. Second Normal Form

A connection is deemed to be in 2nd Normal Form, or 2NF, in DBMS when it is in 1NF but lacks any non-prime assign useable naïve on any candidate key's appropriate subset.

A relation can only exist in the first regular form; it cannot exist in the second normal form at all through partial dependence. The relationship is in the 2NF state if there is no partial dependence between any acceptable subset of any candidate key in the table and any non-prime key characteristic (i.e., traits not contained in any candidate key). When the correct subset of a candidate key determines a non-prime attribute, this is known as partial reliance.

3. Third Normal Form

By applying normalization principles, the Third Normal Form (3NF) is a relational database schema design method that reduces data duplication, stops data abnormalities, preserves referential integrity, and simplifies data management.

A relation is in third normal form if it is in second normal form and there is no transitive dependency for non-prime features. If at least one of the following can account for every non-trivial function dependency in a relation, then the relation is in 3NF.

X –> Y

The super key X is used.

A prime attribute is Y. Every element of Y can be found in a possible key.

4. Fourth Normal Form

The state in which a relation is in Boyce Codd Normal Form and has no multi-valued dependents is known as Fourth Normal Form, or 4NF in DBMS jargon. The relation A –> B has a multi-valued dependency when there are several values of B for a single value of A.
If a relation R is in the Third Normal Form and LHS is the super key for each FD, it is said to be in BCNF.
The relation is in the BCNF if X is a super key in each non-trivial functional dependency X -> Y.

5. **Fifth Normal Form**

Fifth normal form (5NF), also known as projection-join normal form (PJ/NF). It is a database normalization level used in relational databases that store multi-valued characteristics. It removes redundancy by separating several relationships that are semantically connected.

There is a relation in 5NF. If it is in 4NF with no join dependencies, joining should be lossless.

2.3.2 Advantages of Normalization

Normalization helps to remove redundant data in an efficient way.
• A better general organization of the database.
• The internal data consistency of the database.
• A database design that is far more flexible; the concept of relational integrity is upheld.

2.3.3 Disadvantages of Normalisation

- You need to determine the user's needs before building the database
- The performance degrades as the relations are normalized to higher normal forms, like 4NF and 5NF.
- Restoring higher-degree connections to normalcy is a difficult and time-consuming task.

- Careless decomposition can result in a poor database design, which can cause serious issues.

2.4 Decomposition

When a table is split up into multiple tables, the DBMS decomposition technique assists in removing anomalies, inconsistencies, and redundancies from the database. The process of disassembling a relation X into smaller components, as X1, X2, and so forth, is known as decomposition. Decomposition preserves dependently and without loss.

The process of disassembling a table in a database into its component pieces is known as decomposition. Decomposition, then, substitutes a set of multiple smaller relations for a given connection.

The process of breaking down a database table into its component parts is called decomposition. Hence, decomposition substitutes a collection of multiple smaller relations for a single relation. Therefore, each table in a database can be divided into many tables to collect a particular collection of data. It is always necessary to apply lossless decomposition. By employing the decomposed relations, We can guarantee that the correct reconstruction of the data or information from the original relationship is possible. When relationships aren't broken down properly, problems like information loss can eventually occur.

2.4.1 Types of Decomposition

There are two types of Decomposition

1.Lossless Decomposition

A decomposition is considered lossless when joins from the decomposed tables allow one to reconstruct the original relation R. It's the choice that people prefer the most. In this way, the knowledge remains intact even after we break down the relationship. In the end, a lossless join would yield a result that was extremely close to the original relation.

For instance,

Think of "A" as the relational schema containing an instance of "a." Think about how it breaks down into: A1, A2, A3,.... An; for example: a1, a2, a3,... an 'Lossless Join Decomposition' is the term used if a1 ≈ a2 ⋈ a3....≈ an..

2.Lossy Decomposition

When we split a relation into many relational schemas, as the term implies, we inevitably lose data or information when trying to recreate the original relation.

The following qualities must be present for decomposition:

1.Lossless Decomposition Is Required

Information from a decomposed relation must never be lost, so decomposition must always be lossless. By doing this, we are guaranteed that the join will finally produce the same relation as its deconstructed version when the relations are combined.

2. Maintenance of Dependence

Dependency is a crucial restriction on a database; each dependency must be satisfied by at least one deconstructed table. Functional dependence between the two sets is required if $\{P \rightarrow Q\}$ is true. Therefore, if both of these are set in the same relation, it becomes quite helpful when checking the dependency. Only when the functional dependency is maintained can we use this decomposition property. Furthermore, we can examine different updates with this attribute without having to compute the natural join of the database structure.

3. Insufficient Data Redundancy

This property states that redundancy in the data cannot affect decomposition. Careless deconstruction may have negative effects on the database's total data. We can easily attain the property of no redundant data when we execute normalization.

2.5 Synthesis Approach

The synthesis approach is based on the idea that a set of functional dependencies and a collection of attributes can create a database. 3NF or BCNF relations, or fragments thereof, are then synthesised based on the set of dependencies.

- **Knowledge Check 1**

  Fill in the Blanks

1. For a relation to be in ________, it must be in first normal form and cannot have any partial dependencies.

2. If a relation R is in Third Normal Form and LHS is the super key for each FD, it is said to be in ________.

3. ____________ explains how to break down a database table into its component parts.

- **Outcome-Based Activities 1**
  List all the types of normalisation with example.

2.63NF and BCNF

No non-prime attribute can be transitively dependent on the relation's candidate key, according to 3NF.A relation is in the third normal form (3NF) if it is in the second normal form (2NF) and there is no transitive dependency, that is, no non-key attribute that depends on the primary key. It must also meet one of the prerequisites listed below. The C->D function dependence should be a component of the candidate key, and C should be a super key and prime attribute. To reduce data duplication and ensure data integrity, 3NF is utilized.

The candidate keys will be : {LNO, MNO, NOP}

as the closure of LNO = {L, M, N, O, P}

closure of MNO = {L, M, N, O, P}

closure of NOP = {L, M, N, O, P}

Since this relation is already in 2NF and is not transitive, it is in 3NF.Furthermore, no non-prime attribute comes from a non-prime attribute.
But according to BCNF, if a relation has a minimal functional dependency (X -> Y), then X must be a super key. It is possible to construct 3NF and preserve the reliance in the connection. Boyce-Codd normal form (BCNF) was created in 1974 by R.F. Boyce and E.F. Codd. Any of the following could be a sign that a functional reliance is present in BCNF:
It should be in 3NF by now and be a super key for a functional dependency such as P->Q.The restrictions of BCNF, a 3NF offshoot, are stricter than those of 3NF.Moreover, it is said to be more potent than

For example:

In the instance of the functional dependencies A->B, A->C, C->D, and C->A in the relationship R (A, B, C, and D):

As the closure of A = {A, B, C, D}, the potential keys are {A, C}.

C = {A, B, C, D} closure

As the link is already in 3Nf (no prime attribute derives from no prime attribute) and there is a candidate key on the left side of the functional dependency, it is in BCNF.

2.7 Lossless Join and Dependency Preserving Decomposition

In the case when F is a collection of FDs over R and R is a relational schema, a lossless-join is defined as a decomposition of R into two schemas with the attribute sets X & Y. This is due to the fact that every instance of R that meets the dependencies in F will result in instance r from a natural join of X & Y.

A relation can be divided into two or more relations using the lossless-join decomposition technique. This feature ensures that neither the extra-tuple creation issue nor the loss of information from the original relation during the decomposition occur.

Every reliance in the dependency preservation needs to be satisfied by at least one deconstructed table.

When a relation R is split into relation R1 and relation R2, its dependents must come from the union of relation R1 and relation R2's functional dependencies, or they must be a part of relation R1 or relation R2.

2.84NF and Multi valued dependency

A relation in Boyce Codd Normal Form without any multi-valued dependents is called "4NF" in relational database management systems.

When one value of A is given numerous values of B, the relation A > B is said to have multi-valued dependency.

A table is said to have multi-valued dependency (MVD) if many rows in it. It implies that there are many more rows in the same table as a result. Therefore, if there was a multi-valued dependency, the 4NF would not work. At least three table characteristics would be involved in any multi-valued dependency. When two distinct attributes in a given table just so happen to be independent of one another, it's called multi-valued dependency.

However, a third quality is necessary for all of these. At least two of the qualities in the multi-valued dependence are dependent on the third property. It always has at least three of the qualities because of this.

According to an MVD, there can be variations in the attribute "y" values for various values of the variable "x."

So, this is how we'll format it:

x –> –> y

If every pairing of tuples p1 and p2 in q has a legal relation, q(Q), such that one attribute, let's say x, can define another attribute, let's say y.

p1[a] + p2[a]

Then, p3 and p4 exist in r in such a way that.

p1[a] + p2[a] + p3[a] + p4[a]

p1[b] = p3[b]; p2[b] = p4[b]

p1 = p4 and p2 = p3.

ence, MVD is present in this case

2.9 5NF and Join dependency

The project-join normal form is another name for the Fifth Normal Form, or 5NF.If a relation is in 4NF and does not have lossless decomposition into smaller tables, it is in Fifth Normal Form (5NF).If the candidate key reveals each join dependence in the relationship, it is also possible to assume that the relation is in 5NF.. The idea of join dependency is directly based on the Fifth Normal Form, or 5NF for short. Just like functional or multi-valued dependencies, join dependencies are limitations. It can only be fulfilled in the event that the relevant relation connects a set of projections. Join dependence, or JD for short, is a type of constraint that is similar to functional dependency (FD) or multi-valued dependency (MVD). Only in situations where the relationship.

If a table can be replicated by just linking numerous other tables, each of which has a subset of the original table's attributes, then that table is deemed to be a join dependence.Therefore, it is similar to an MVD generalisation. The JD and 5NF have a connection. A relation can only be in the 5NF in this case if it is also in the 4NF.Keep in mind that it cannot degrade any further.

- **Knowledge Check 2**

  State true and false for the following sentences.

  - 1. If a table can be replicated by only joining numerous new tables, each of which has a subset of the original table's characteristics, then the table is said to be multi-valued dependent.
  - 2. In relational database administration systems, a relation in Boyce Codd Normal Form without any Join-valued dependents is referred to as "4NF".

.

- **Outcome-Based Activities 2**
  List some examples of Multi-valued dependency and Join dependency.

2.10 Summary

- A relationship between two attributes is called a functional dependency (FD); in a database, this relationship is typically between the main key (PK) and other non-key attributes. Attribute Y in any relation R is functionally reliant on attribute X (often the PK) if the value of attribute X influences the value of Y in a different way for each valid instance of X.
- The entirety of all qualities from which a functional reliance may be functionally inferred using Armstrong's criteria for inference is known as the closure of that functional dependency. If "F" is a functional dependency, then "F+" can be used to indicate the closure of that functional reliance.
- The process of structuring the data in a database is called normalization. Redundancy in a relationship or set of relationships is eliminated by normalization. Eliminating undesired characteristics like Insertion, Updating, and Deletion Anomalies is another cause.
- Data in a database is arranged via the normalization process. Tables must be built and their associations defined by rules in order to guarantee data security and enhance database flexibility by removing redundancy and inconsistent reliance.
- In multi-valued dependency, several distinct values are associated with a determinant. When separated, the MVD shows no anomalies related to alteration.
- Join dependence (JD) is a type of constraint that is similar to functional dependency (FD) or multi-valued dependency (MVD).JD is only satisfied when the relation in question is a join of a specific number of projections. As a result, this type of restriction is known as a join dependency.
- An association between two attributes, most frequently the PK and other non-key properties in a database, is called a functional dependency (FD).If value of property X (often the PK) affects value of attribute Y in a different way for every valid instance of X, then attribute Y is functionally dependent on attribute X for each relation R.

2.11 Self-Assessment Questions

1. What is Functional Dependencies and its implications?
2. What is Closure rules?
3. Explain Normalisation.
4. What is Decomposition?
5. What is the Synthesis approach?
6. Explain 3NF and BCNF.
7. What is Lossless Join and Dependency Preserving?
8. What is Join Dependency?

2.12 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 3

RELATIONAL DATABASE MODEL

Learning Outcomes:

- Students will be capable of learning about the Relational Data Models
- Pupils will be able to comprehend relational database limitations and concepts.
- Using ER, students will be able to comprehend database design concepts.
- Students will be able to understand about the EER to Relational Mapping and Relational Algebra.

Structure

3.1 Relational Data Models

The relational version (RM) represents the database as a set of members of the family. A values desk is all that a relation is. The desk`s rows are collections of pertinent statistics values. Each row withinside the desk represents a actual person, place, object, or connection. Using the desk call and column names enables information the which means of the values in every row. To constitute the statistics, a hard and fast of members of the family is employed. Tables are used withinside the relational version to shop statistics. However, there's no connection among the bodily garage of the statistics and its logical arrangement.

The relational version for database control is a technique for dealing with and conceptually describing the statistics saved in a database. The facts on this version is prepared into a set of two-dimensional tables, or members of the family, which can be jointly exclusive. The relational version, proposed via way of means of E.F. Codd, makes use of tables or members of the family to symbolize statistics.

The conceptual database version need to be created the usage of an ER diagram, after which it need to be transformed right into a relational version that may be carried out with any RDBMS language, such MySQL, Oracle SQL, and so on. The relational version governs how statistics is saved in relational databases. Data is stored in relational databases as members of the family, or tables.

Think approximately the relation STUDENT, which has the Table 1 attributes ROLL NO, NAME, ADDRESS, AGE

ROLL NO	**NAME**	**ADDRESS**	**AGE**
1	RAHUL	RAIPUR	25
2	KIRTI	BILASPUR	22
3	SWATI	BHILAI	21
4	ROHAN	JAMSHEDPUR	24

Some of the famous Relational Database Management System are:

- DB2 and Informix Dynamic Server via way of means of IBM
- Oracle and RDB via way of means of Oracle

- SQL Server and Access via way of means of Microsoft

3.1.1 Characteristics of Relational Model

- Data is proven as rows and columns, or relations.
- The relational version refers to the connection among the tables that shop statistics.
- Data definition, statistics manipulation, and transaction management are made less difficult via way of means of the relational paradigm.
- Every column represents an characteristic and has a awesome name.
- The unmarried entity is represented via way of means of every row.
- The relation's name is unique compared to all other relations.
- There is exactly one atomic (single) value in each relation cell.
- Each characteristic has a unique name.
- The attribute domain is meaningless.
- The values in a tuple are unique.
- Tuple order may take on different sequences.

3.1.2 Important terms related to the Relational Database

- A desk`s columns are all attributes. Attributes are the functions that symbolize a relation.
- Student Roll No., Name, NAME, etc.
- Tables: In the relational paradigm, family members are saved in a desk format. It is saved collectively with its entities. A desk includes parts: rows and columns. Rows constitute records, and columns mirror features. .
- A unmarried desk row inclusive of a unmarried report is referred to as a triple.
- Relation Schema: A relation schema represents the call of the relation in addition to its attributes.
- Degree: The whole variety of features that incorporate the connection is its degree.
- The overall variety of rows, or cardinality, withinside the desk.
- Column: The column presentations an characteristic's variety of values.
- In an RDBMS system, a relation example is a confined set of tuples.
- Relation items by no means include reproduction tuples.
- The relation secret is the set of one, , or greater features which might be found in each row.

- An characteristic's characteristic area is the predetermined price and scope that every characteristic possesses.

3.2 Concepts and Relational database constraints

Constraints are the regulations which can be positioned at the statistics columns of a desk. To restrict the sorts of statistics that may be entered into tables, filters are used. This guarantees the accuracy and reliability of the statistics within side the database. Constraints on the desk and column ranges are each feasible.

3.2.1 Relational Model Constraints

The situations that have to be happy with the aid of using database statistics so as for the Relational version to be legitimate are referred to as constraints. These regulations are checked out earlier than to doing any database operations (which includes insertion, deletion, and updating). If any of the regulations are breached, the technique will fail. In DBMSs, integrity regulations can take many exceptional forms.

The 3 essential classes of regulations at the relational database control machine are:

- Domain Constraints
- Key Constraints
- Entity Integrity Constraints
- Referential Integrity Constraints

Domain Constraints

Domain constraints can be violated if an characteristic cost is incorrectly statistics-typed or does now no longer arise within side the connected area. Every characteristic have to have a completely unique cost for each tuple, in keeping with area requirements. Examples of this encompass characters, Booleans, actual numbers, integers, and variable period strings, amongst different famous statistics types.

For example

Create DOMAIN Customer Name

CHECK (value not NULL)

For example

Eid	**Ename**	**Phoneno**
1	Rohan	9994678990,9690578999

The domain requirement is violated as the Phone is a multi-valued attribute and the Name in the connection is a composite attribute.

Key Constraints

All relation in the DBMS should have single set of attributes that define a tuple uniquely. That group of characteristics is referred to as keys. A key in the context of a student is ROLL NO. Two students cannot have the same roll number. Because they ensure that every tuple in the relation is unique, they are referred to as uniqueness constraints.

There may be multiple keys in a relation, or candidate keys (minimum super keys), from which we choose one to be the primary key. The candidate key with the fewest attributes should be chosen, but there are no limitations on the candidate key we choose as the main key. Just as the primary key cannot contain null values, the key also needs to meet the Not Null constraint.

The following are the two characteristics of a key:

- For each tuple, it must be different.
- It cannot contain any NULL values.

For Example

EID	**ENAME**	**AGE**
1	Rahul	22
2	Rohan	21
3	Rohit	23
4	Amar	21
1	Armaan	20

The key constraint is broken in the given table since the first and last tuples have the identical EID value of 01

.

Entity Integrity Constraints:

No primary key can have a NULL value, according to entity integrity constraints, because we need primary keys to uniquely identify each tuple in a relation.

EID	**ENAME**	**AGE**
1	Rahul	20
2	Rohan	21
44	Karan	18
NULL	Prem	23

In the relationship mentioned above, EID is designated as the primary key. As the primary key cannot contain NULL values, it is violated in the third tuple.

Referential Integrity Constraints

Referential integrity occurs when an attribute of a connection can only accept values from that relation or from the same relation. Foreign keys serve as the foundation for referential integrity constraints in database management systems. A foreign key is a crucial component of a relation and should be discussed in connection to other relations.. A crucial feature of an identical or different connection is mentioned in a relational integrity constraint state. However, that crucial element needs to be present in the table.

EID	**ENAME**	**DNO**
1	Rohan	12
2	Rohit	22
4	Amit	14

Dno	**Place**
12	Jaipur
13	Bhilai
14	Raipur

In the aforementioned tables, the DNO of Table 1 is the foreign key and the DNO of Table 2 is the main key.Since DNO = 22 is not defined in Table 2's primary key, it cannot be utilized in Table 1's foreign key. Thus, in this case, referential integrity restrictions are violated..

Features of Relational Database Model –

The relational database features are discussed briefly.

1 – Simplicity of Model

The relational database version is a long way much less complex than different varieties of database models. Simple SQL queries are enough to deal with the information as it lacks question processing and structuring, negating the want for complex queries.

2 – Ease of Use

Without turning into slowed down withinside the database`s complexity, customers can fast and surely reap the records they want. Complex question operations are achieved the usage of Structured Query Language (SQL).

3 – Accuracy

Relational databases are characterised with the aid of using their tight definition and organization, which prevents information duplication. Because in their established information and shortage of reproduction entries, relational databases are accurate.

4 – Data Integrity

Because RDBMS databases provide consistency throughout all tables, they're additionally often hired for information integrity. Features like accuracy and user-friendliness are assured with the aid of using the information integrity.

5 – Normalization

Effective strategies of storing records are getting increasingly vital because it grows in complexity. Normalization is a way used to lessen garage area with the aid of using dividing facts into small, digestible bits. Data may be prepared into levels, and every stage calls for practice earlier than intending to the following stage of records normalization.

Additionally, database normalization ensures that a relational database`s shape is uniform and unvarying and that it can be as it should be changed. This ensures the integrity of the records you operate on this database to make enterprise choices.

6 – Collaboration

Even whilst records is being updated, numerous customers can nonetheless get entry to the database and reap facts simultaneously.

7 – Security

The Relational Database Management System guarantees records safety with the aid of using proscribing direct get entry to to legal customers only. The facts isn't on hand to unauthorized customers.

3.2.2 Advantages of the Relational Model

- It is easier to understand and much easier than both the network and the hierarchical model.
- The relational model's structure can be altered to meet specific needs.
- It's Modular
- Reliability of data
- It ia safety
- Data reliability
- Applications for operations are simple.

3.2.3 Disadvantages of the Relational Model

- Unsuitable for huge databases
- Relationships between tables can occasionally be challenging.
- It is very complex and hard to use.
- As the quantity of statistics grows, relational database protection receives more difficult over time. The protection of the database calls for a big quantity of labor from builders and programmers.
- The setup and protection of a relational database gadget are expensive.
- The overall performance of the database will go through because it grows larger or extra dispersed over extra servers because of troubles with availability and latency.
- Because relational databases can also additionally most effective preserve tabular statistics, it is probably difficult to depict complex relationships among special objects. This is a hassle due to the fact a whole lot of packages want a couple of tables to preserve all of the statistics that their utility good judgment calls for

- The relational database can also additionally turn out to be slower for motives apart from simply the usage of several tables. The complexity of the gadget will increase while there are extra tables and statistics in it. Depending at the variety of customers logged in at any given moment, it may bring about behind schedule question reaction instances or maybe a complete failure for them.

3.2.4 Operations in Relational Model

The relational database model supports the following four fundamental update operations:

• The insert command is used to add data to the relation.

• Use delete to get rid of tuples from a table.

• The values of some of an existing tuple's attributes can be changed.

• Select lets you choose a specific collection of facts.

• The insert command is used to add data to the relation.

• Use delete to get rid of tuples from a table.

• The values of some of an existing tuple's attributes can be changed.

• Select lets you choose a specific collection of facts.

- **Knowledge Check 1**

  Fill in the blanks with the following sentences.

1. The basis for DBMS referential integrity constraints is Foreign Keys.
2. ___________ happens when a relation's attribute can only accept values from that relation's attribute or from any other relation.

3.3 Database design using ER

An entity-courting version (ER version) is called such. This is a high-stage records version. This version is used to specify the records gadgets and relationships for a selected system. It develops the conceptual layout of the database. It additionally produces a completely easy and clear-cut records view. In entity-courting modeling, an entity-courting diagram represents the database structure.

A diagram that illustrates the connections among entity units which might be stored in a database is referred to as an entity courting diagram, or ER Diagram for short. Stated differently, ER diagrams assist describe the manner databases are logically organized. The 3 center ideas that shape the idea of an ER diagram are entities, characteristics, and relationships. An powerful manner to visualize the ER paradigm is thru ER diagrams. Peter Chen proposed the ER Diagram in 1971 with the purpose of making a uniform conference that would be carried out to relational databases and networks. His plan changed into to apply an ER version as the idea for a conceptual modeling method. In ER Diagrams, entities are represented with the aid of using rectangles, attributes are proven with the aid of using ovals, and relationships are proven with the aid of using diamond shapes.

Consider the introduction of a faculty database, for example. In this database, the pupil may be an entity with facts approximately them, together with their name, residence, ID, age, and different facts. It is viable to consider the city, road name, pin code, and different information of the deal with as wonderful matters which might be related to the deal with.

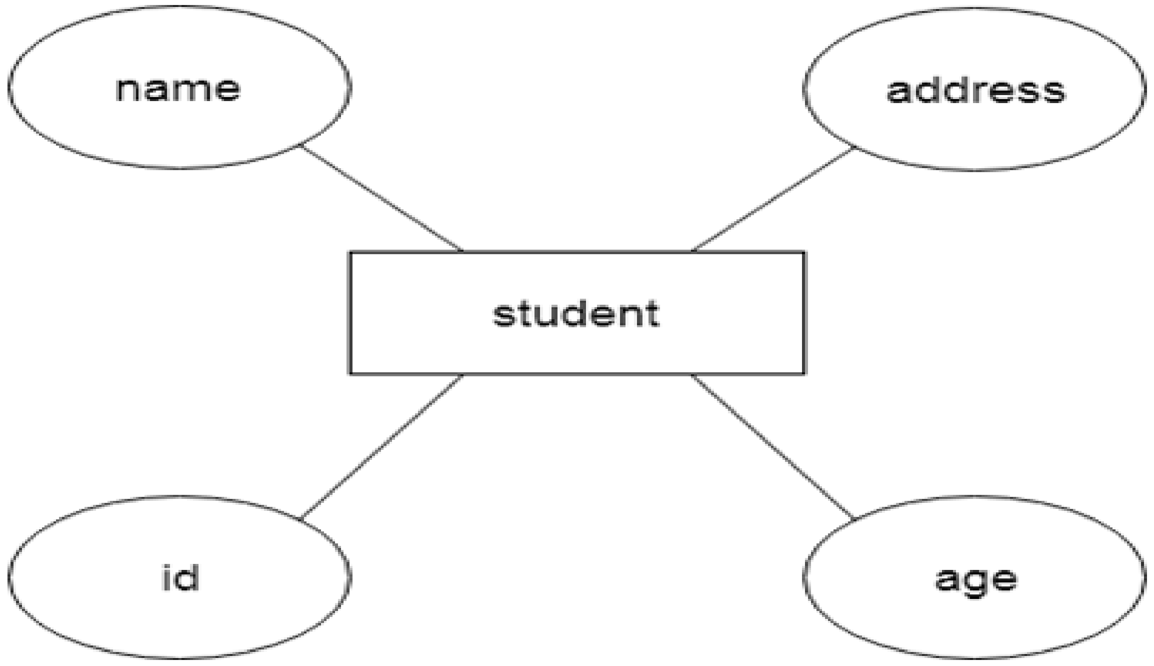
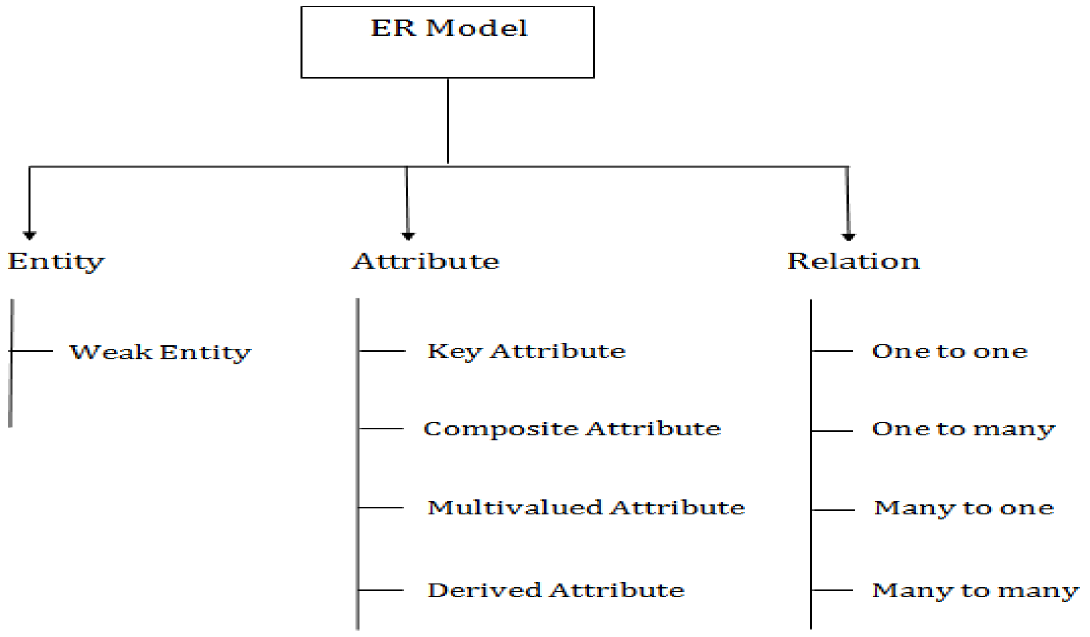


Fig. 3.1 **Components of the ER Model**



3.2 ER Diagram Symbols and Notations

The rectangle, oval, and diamond are the three primary symbols used in entity relationship diagrams to show the relationships between items, entities, and attributes. The primary components of the ERD Diagram serve as the foundation for many sub-components. Using a range of ERD symbols and notations, an ER Diagram is a visual representation of data that shows how data are related to one another.

Squares: Entity relationship diagrams utilize this entity-type symbol.
Ellipses: Attribute symbols
Diamonds: This symbol represents different types of relationships.
Lines: It links attributes to entity types and entity types to other relationship types.
Principal key: highlighted attributes
Two Ellipses: Show attributes that have several values.

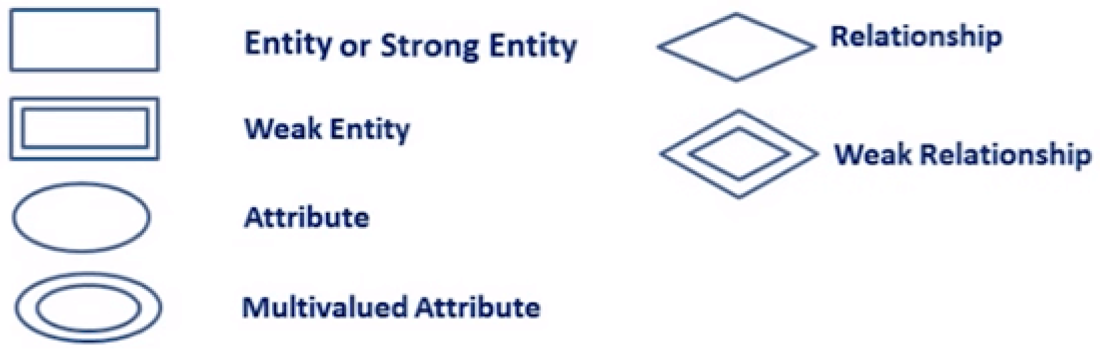


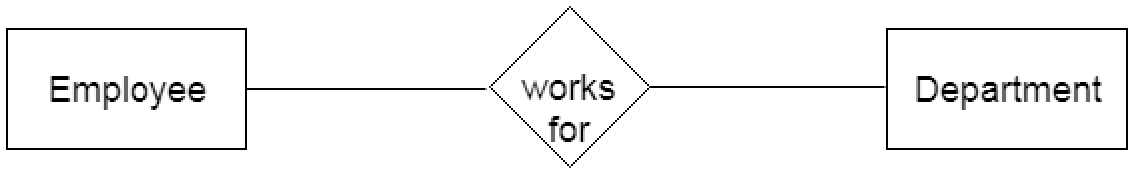
Fig. 3.3.1 Components of the ER Diagram

This model is basically based on three basic concepts:

- Entities
- Attributes
- Relationships

Entities

Any item, group, person, or location can be considered an entity. Rectangles can be used to symbolize different entities in the ER diagram. We can observe that a manager, a product, an employee, a department, etc. are all different things by using a corporation as an example.



Dependency on another entity is what makes an entity weak. There isn't a single distinguishing feature on the weak thing. A double rectangle serves as the weak entity's representation.

Attributes

It is a single-valued property of the relationship or entity types. For example, a lecture may have the following details: date, time, length, venue, etc. An entity's attribute describes its property. Eclipses represent certain qualities.

Types of Attributes

1.Key Attribute

The primary feature of an entity is represented by its key attribute. It stands for a primary key. The text is underlined and an ellipse represents the significant feature.Simple qualities cannot be further classified. For instance, a student's phone number. An atomic value is another word for it.

2. Composite Attribute

An characteristic this is composite is one this is composed of a couple of different properties. The composite characteristic is represented through an ellipse, and in addition ellipses are applied to sign up for the ones ellipses. Name, for example, includes the first, middle, and closing names.

3.Multi-valued Attribute

An characteristic may have multiple cost given to it. We name them traits with a couple of values. A double oval represents a multi-valued characteristic. For example, a scholar can also additionally have multiple tele cell smart phone number.

4.Derived Attribute

An characteristic that may be constituted of any other characteristic is referred to as a derived characteristic. You can also additionally use a dashed ellipse to symbolize it. For example. Date of Birth.

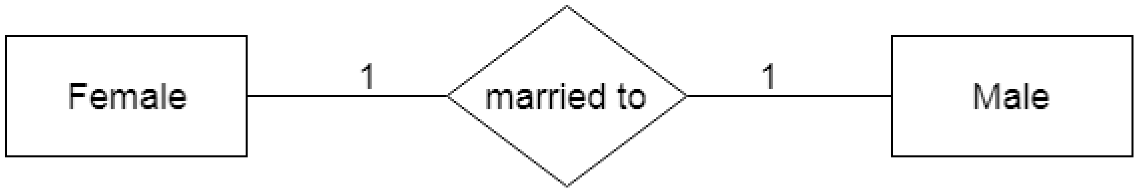
Relationships

- A dating is the connection among entities. Diamonds or rhombuses are applied as symbols of the partnership.

The following lists many cardinal connection types:

One-to-One Relationships

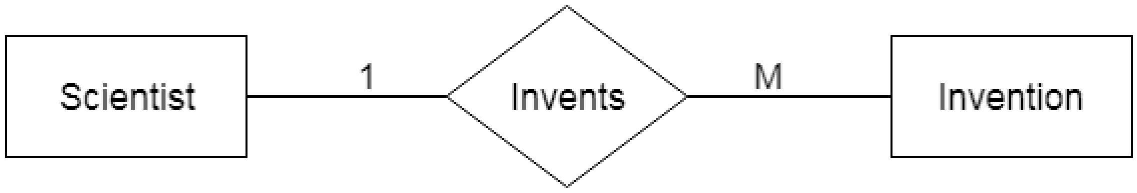
Relationships that contain best one example of every entity are called one-to-one relationships.
For example, a lady can also additionally marry one man, and a person can also additionally marry one lady**.**



One-to-Many Relationships

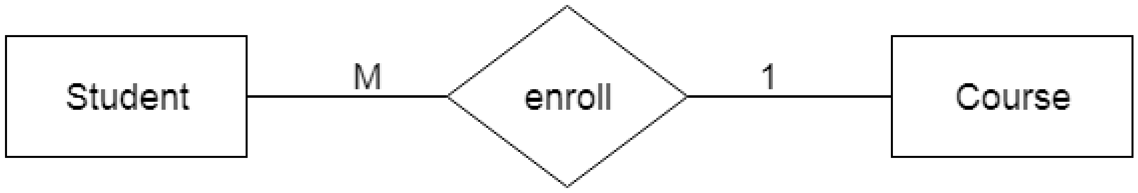
When an entity has numerous instances connected to the relationship on the right and just one instance on the left, this is known as a one-to-many relationship.

For example, several ideas can be conceived of by a scientist, but only that one scientist can really make them.

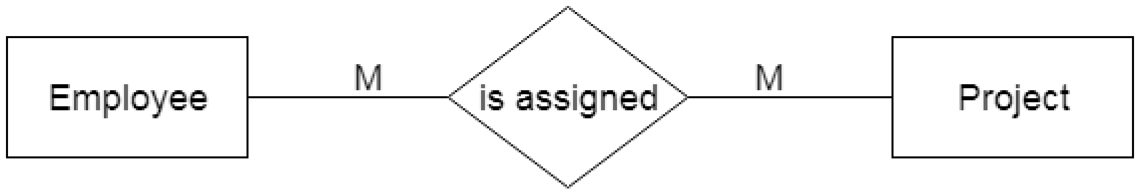


Many to One Relationships

When a relationship is linked to multiple instances of the left-hand entity and only one instance of the right-hand entity, it is referred to as a many-to-one connection. For instance, even though a student only enrolls in one course, the course may have multiple students.



- **Many-to-Many Relationships**
    - •A relationship is referred to as many-to-many when there are multiple instances in which the entities on the left and the right are connected to it.
      Employees might be allocated to several projects, for instance, and vice versa.



3.3.2 Create an Entity Relationship Diagram (ERD)

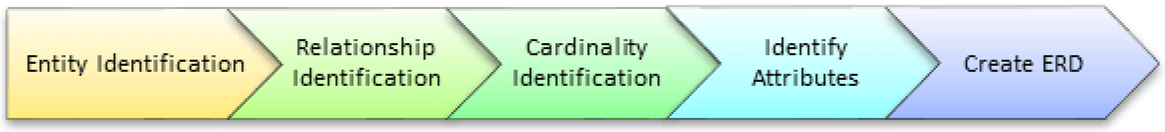


Fig.3.3.3 How to create ERD

- It offers you definitions for phrases utilized in entity courting modeling. • It helps you to realize earlier what fields every desk could have and the way your tables ought to hyperlink to 1 another.
- Explains relationships, entities, and properties.
- Database designers use ER diagrams to plot out the implementation of information in unique software program applications, which quickens the advent of databases. ER diagrams may be was relational tables.
- The database fashion dressmaker can higher recognize the information as a way to be saved within side the database through the use of ERP diagrams.
- ERD Diagram allows customers to engage with the logical shape of the database.
- The ER version is a famous graphical device for information modeling and a GUI illustration of the logical shape of a database. It can be used to assemble database designs.
- It helps the identity of the entities and their connections inside a system.

3.4 ER to Relational Mapping and Relational Algebra

3.4.1 ER to Relational Mapping

The ER Model offers a better, easier-to-understand overview of entity-relationship when it is represented as diagrams. ER diagrams can be mapped to relational schema, meaning that relational schema can be generated using them. We can create an approximate schema, but we can't import every ER constraint into the relational model.

The following are the general steps involved in converting an E-R diagram into a relational model:

- Mapping of an entity set into database relations (tables).
- The attributes of an entity are among the attributes of a table.
- The main key of the relation is the key attribute of an entity.

ER-to-Relational Mapping Algorithm

1. Step 1: Mapping of Regular Entity Types.
2. Step 2: Mapping of Weak Entity Types.
3. Step 3: Mapping of Binary 1:1 Relation Types.
4. Step 4: Mapping of Binary 1: N Relationship Types.
5. Step 5: Mapping of Binary M:N Relationship Types.
6. Step 6: Mapping of Multi-valued attributes.

Mapping Process

• Make a weak entity set table.
• Add each of its properties as a field to the table.
• Include the identifying entity set's primary key.

• List all restrictions on foreign keys.
• Make tables for every higher-level entity.
• Create tables for entities at a lower level.
•Add the main keys of the higher-level entities to the lower-level entities' database.

• Add all additional properties from lower-level entities to lower-level tables.
• Indicate which is the lower-level table's main key and the higher-level table's primary key. Set restrictions about the foreign keys.

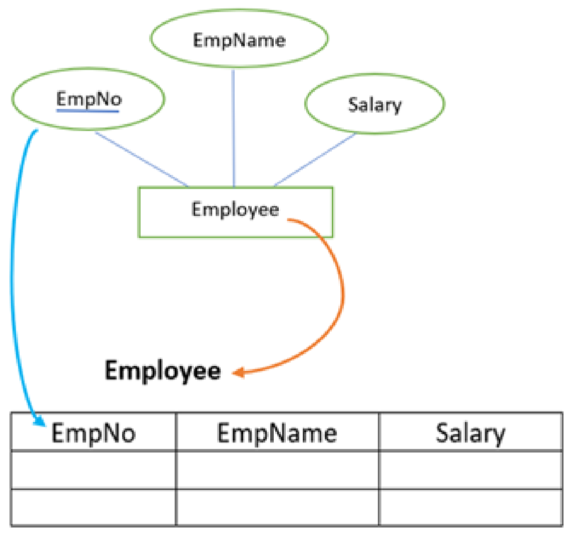


Fig. 3.4.1 Storing data in table using ER-Diagram

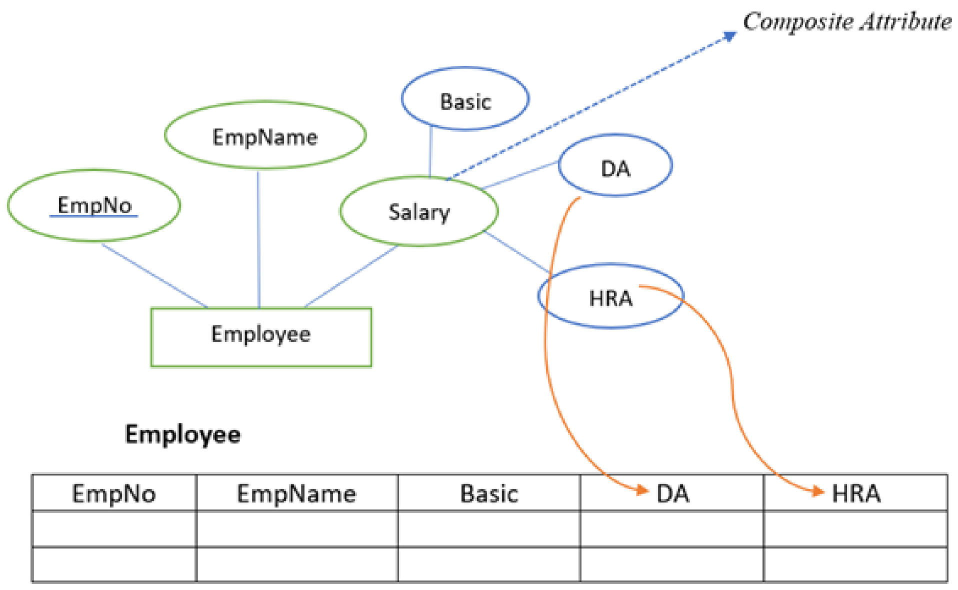


Fig. 3.4.2 Relational algebra

Relational algebra is a procedural question language that takes members of the family times as enter and produces members of the family times as output. It makes use of operators to execute queries. Either a unary or binary operator may be used. They well known that their consumption and output are members of the family. Intermediate effects which are additionally taken into consideration members of the family are received with the aid of using recursively making use of relational algebra to a relation.

The essential operations of relational algebra are given underneath as follows −

- Select
- Project
- Union
- Set exclusive
- Cartesian product
- Rename

1. Select Operation (σ)

It selects tuples that fulfill the given predicate from a relation.
Symbol− σp(r)
Where σ stands for the choice predicate, and r stands for relation. p is prepositional common sense formulation which may also use connectors like and, or, and now no longer. These phrases may also use relational operators like − =, ≠, ≥, < , >, ≤.
For example −

σsubject = "database"(Books)

2.Project Operation (∏)

It initiatives column(s) that fulfill a given predicate.

Notation − ∏A1, A2, An (r)

Where A1, A2 , An are characteristic names of relation r.

Duplicate rows are mechanically eliminated, as relation is a set.

For example −

∏subject, author (Books)

3.Union Operation (∪)

It plays binary union among given members of the family and is described as −

r ∪ s = t ∈ r or t ∈ s

4. Set Difference (−)

Tuples which are found in one relation however absent from the second one are the final results of the set distinction operation.

Notation − r − s

Finds all of the tuples which are found in r however now no longer in s.

∏ author (Books) − ∏ author (Articles)

5. Cartesian Product (X)

Combines facts of exclusive members of the family into one.

Notation − r X s

Where r and s are members of the family, and their output can be described as −

r X s = q ∈ r and t ∈ s

σauthor = `tutorialspoint'(Books X Articles)

6. Rename operation (ρ)

Relational algebra produces family members, however those family members don`t have names. The rename operation may be used to rename the output relation. The "rename" system is denoted with the aid of using the small Greek image rho.

Notation − ρx (E)

Where the end result of expression E is stored with call of x.

Additional operations are −

- Set intersection
- Assignment
- Natural join

Knowledge Check 2

State True and False for the Following Sentences.

1. An characteristic that may be constructed from any other characteristic is referred to as a composite characteristic. You might also additionally use a dashed ellipse to symbolize it.
2. When there are various times of the entity at the left and the entity at the proper which can be linked to the connection, the connection is stated to be many-to-many.
3. An characteristic could have multiple fee given to it. We name those "derived attributes."

Outcome-Based Activity 2

List out all of the steps of ER to Relational Mapping and Relational Algebra.

3.5 Summary

- The relational version in DBMS is an summary version used to arrange and alter the information saved in a database.It maintains information in two-dimensional related tables, or family members, in which every row represents an item and every column represents one in all its attributes.

- Any item information member kind may be converted right into a corresponding relational database (SQL) information supply illustration in any relational database this is supported with the aid of using a relational mapping. An item version may be mapped right into a relational information version the use of relational mappings.
- Constraints are a method of in addition proscribing information past what the Domains allow.
- Mainly Constraints at the relational database are of five types:
  - Domain constraints.
  - Key constraints.
  - Entity Integrity constraints.
  - Referential integrity constraints.
  - Tuple Uniqueness constraint.
- Instances of relations are accepted as input by relational algebra, which then outputs instances of relations. Procedural algebra is a language for creating queries. It uses operators to execute queries. It is feasible to have both binary and unary operators.

3.6 Self-Assessment Questions

1. What are Relational Data Models?
2. What are Relational Database Constraints?
3. What is ER Diagram?
4. What is database design using ER?
5. What is Relational Mapping?
6. What is Relational algebra?
7. Explain ER to Relational Mapping.
8. Explain ER to Relational algebra.

3.7 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.

- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 4

QUERY PROCESSING

Learning Outcomes:

- Students will be capable of learning about the Basics of Query Processing.
- Learners will comprehend how joins are processed.
- The ideas of Materialized vs. Pipelined Processing will be clear to students.
- Pupils will be able to comprehend ACID properties and database transactions.
- Information regarding the Interleaved Executions and Schedules will be available to students.
- Students will learn about the Serializability.
- Students will understand the concept of Database Recovery and Backup.

Structure

4.1 Introduction to Query Processing

The process of translating high-level inquiries into low-level expressions, which are usually used at the file system's physical level, optimizing the query, and then executing it to get the intended result is known as query processing.

The process of extracting data from a database is called query processing. There are several phases involved in getting data out of the database during query processing.

The following processes are involved:

• Translation and parsing

First, high-level database languages like SQL are used to translate the user queries that have been provided. It is transformed into phrases that can be applied further at the file system's physical level. Following this, the queries are really evaluated together with a number of query-optimizing changes. Thus, a computer system must convert a query into a language that is legible and intelligible by humans before processing it's; the greatest option for humans is SQL, or Structured Query Language. Yet, it is not entirely appropriate for the internal representation of the system's question.

For the internal representation of a query, relational algebra works well. The query parser is analogous to the translation process in query processing. When a user conducts a query, the parser in the system first evaluates the syntax of the query before checking the name of the relation in the database, the tuple, and then the needed attribute value. A tree of the query is created by the parser, or "parse-tree." Also, convert it to relational algebraic form.

As a result, any query that uses views is completely replaced.

- **Optimization**

Several sorts of inquiries have varying costs associated with their examination. Despite the fact that the system is in charge of creating the assessment plan, the user is not required to write their query effectively. A database system often creates a cost-effective query assessment plan to maximize efficiency. Query optimization is the term for the task that the database system does in this manner. The query optimizer should have an estimated cost analysis of each action before it can optimize a query. It's because execution costs, memory allocations to other processes, and other factors all affect how much an operation will ultimately cost.

- **Evaluation**

To do this, in addition to the translation into relational algebra, the criteria for defining and assessing each operation must be annotated on the translated relational algebra expression. As a result, once the system has translated the user question, a query assessment plan is implemented. The question is evaluated by the system, which then generates its output after choosing an assessment strategy.

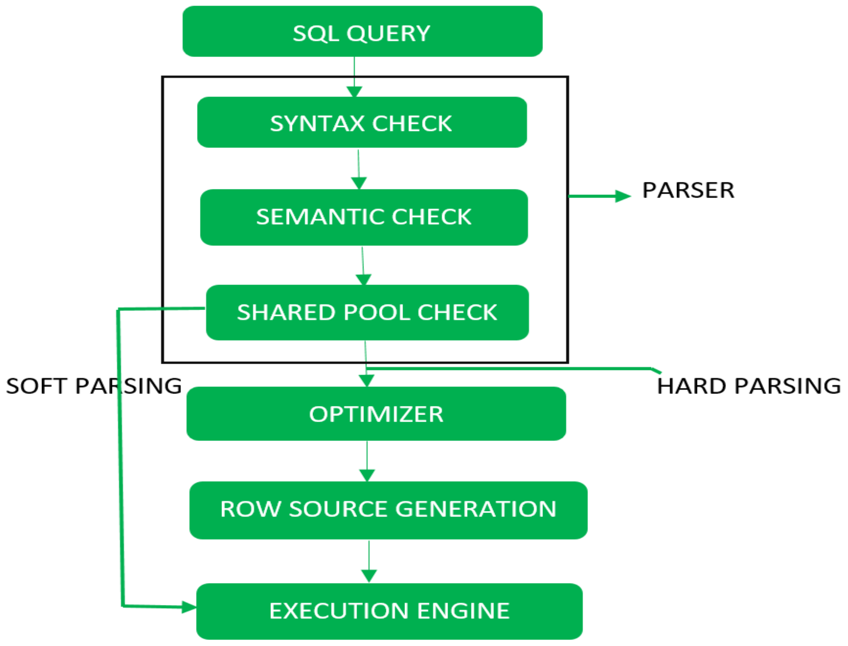


Fig 4.1 Detailed Diagram of query processing

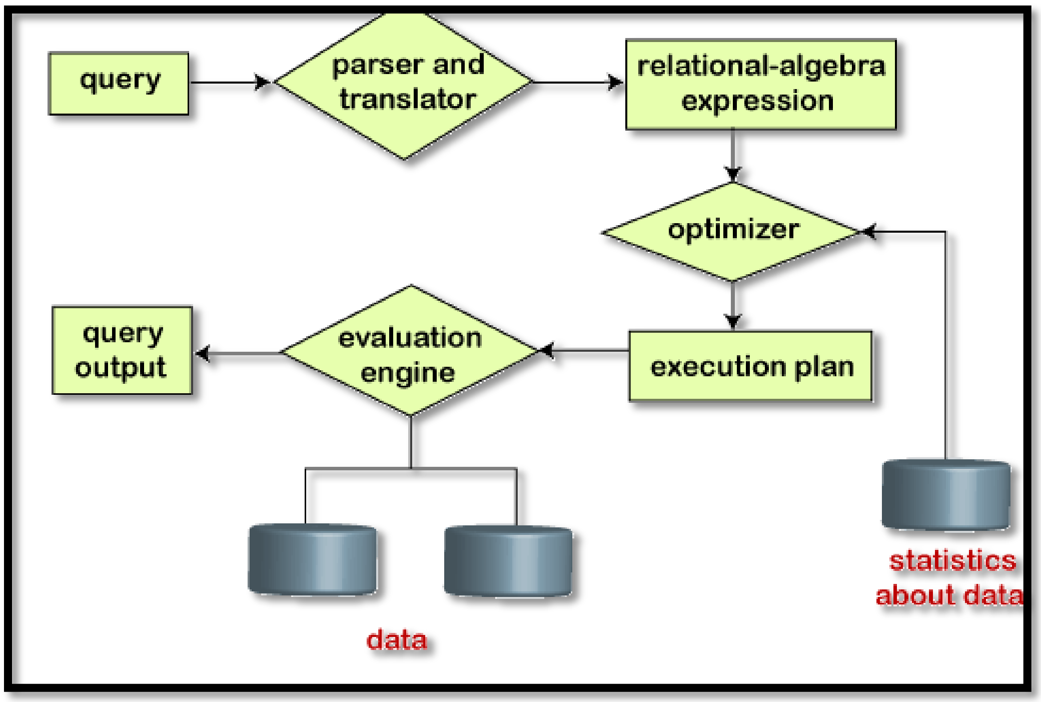


Fig. 4.2 Steps in query processing

4.2 Processing of Joins

A join in a database management system (DBMS) is a binary action that combines selection with a join product in one statement. A join condition is used to allow data from two or more DBMS tables to be combined. In database management systems, tables are linked together using primary keys and foreign keys. Related tuples from different relations are merged via a join operation if and only if a particular join condition is satisfied. The symbol used at the place of join is ⋈.

Employee

EMP_Code	EMP Name
101	John
102	Rahul
103	Rohit

Salary

EMP_CODE	SALARY
101	40000
102	50000
103	35000

Operation: (EMPLOYEE ⋈ SALARY)

EMPCODE	**EMPNAME**	**SALARY**
101	**John**	**40000**
102	**Rahul**	**50000**
***103**	**Rohit**	**35000**

Types of Joins

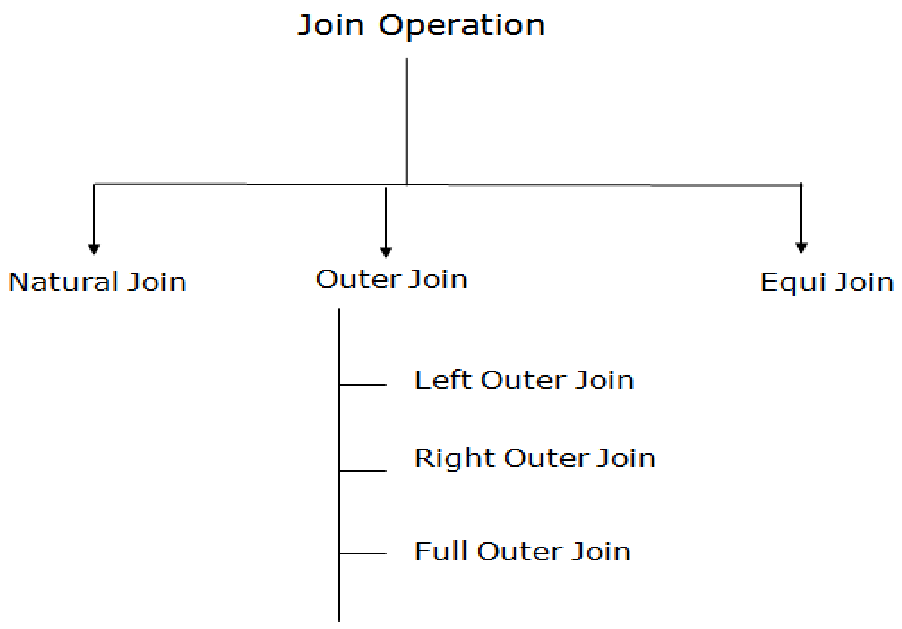


Fig 4.3 operation in join

4.3 Materialized vs Pipelined Processing

Although both methods are utilized for evaluating expressions with numerous operations, they don't differ much from one another.

The table below lists the difference points.

Pipelined Processing	Materialized
It is described as the contemporary method of assessing several procedures.	It is described as the conventional method of assessing several procedures.
The outcomes of the assessed activities are not stored using any temporary relations.	The outcomes of the finished actions are stored in temporary relations. As a result, more I/O and temporary files are needed.
It is a more efficient way of query evaluation due to its quick results creation.	Due to the longer generation time for the query results, it is less efficient than pipelined processing.
Memory buffers are frequently required in order to provide outputs. Insufficient memory buffers will cause thrashing.	It has no more strict memory buffer requirements for query evaluation.
The cost of evaluating queries is maximized. Because reading and writing temporary storages is not covered by the price.	The total cost include the operating expenses as well as the expenses incurred by reading and writing the results to the temporary storage.
Performance problems in the event of trashing.	There is no discarding in materialization. Materialization therefore functions better under these conditions.

4.4 DB transactions

A transaction could be used to describe a collection of tasks. A transaction is an activity that you want to manage "as a whole".

It both takes place in complete or now no longer at all. A unmarried activity is the smallest processing unit that isn't amenable to similarly division. A database transaction is an independent, consistent, and dependable paintings unit this is executed in opposition to a database the use of a database control gadget (or a gadget just like it). A transaction is generally used to explain any database modification. A collection of movements executed on a database which are all finished as a unmarried logical paintings unit, both completely or in part, is called a database transaction.

Stated otherwise, it by no means happens that most effective a subset of the tactics are executed and the effects are retained. While a database transaction is running, the nation of the database might be in brief inconsistent; nonetheless, adjustments are made while the transaction commits or ends.

Let`s take an instance from a easy transaction.
Consider a switch of Rs. 500 made through a financial institution worker from A to B. Several low-degree operations are worried on this extraordinarily small and easy transaction. Transferring cash among financial institution bills is the traditional instance. In order to perform that, you should first withdraw the finances from the supply account after which switch them to the vacation spot account. The manner wishes to be absolutely effective. If you forestall withinside the middle, the cash might be lost, that's Very Bad.
In present day databases, transactions are hired to make certain that records that has been partly written through a person else can't be retrieved. However, the underlying concept continues to be the same: transactions exist to assure that the records you address is preserved beneathneath all circumstances.

4.5 ACID Properties

Small programming gadgets known as transactions can incorporate numerous low-degree operations. In a database gadget, a transaction should meet the standards called atomicity, consistency, isolation, and durability, or ACID properties, so that you can hold accuracy, completeness, and records integrity. A transaction may be used to specific a extrade in nation. The 4 qualities, collectively called the ACID Property, are found in best transactions.

Atomic

You never get to witness "half a change" because once the change is committed, it happens all at once. There is no "half a change"—if the change is committed, it happens all at once.

Consistent

Any attempt to dedicate an invalid alternate will fail, returning the gadget to its earlier legitimate kingdom. The alternate can simplest take location if the brand new kingdom of the gadget might be legitimate. Any try and dedicate an invalid alternate will fail, returning the gadget to its preceding legitimate kingdom. The alternate can simplest take location if the brand new kingdom of the gadget is legitimate.

Isolated

Nothing approximately the transaction is understood to every body else till it's miles finished. It ought to be viable for the database to keep all of its maximum current updates even within side the occasion of a gadget crash or restart. If a transaction updates a few statistics in a database and commits, the database will hold the up to date information. If a transaction commits however the gadget fails earlier than the statistics can be dedicated to the disc, the statistics may be up to date as soon as the gadget resumes.

Durable

If the gadget indicators that the transaction has been dedicated, the customer want now no longer fear approximately "flushing" the gadget to make the alternate "stick" as soon as it has occurred. When numerous transactions are being run concurrently and in parallel in a database gadget, the isolation belongings specifies that every transaction may be treated and finished as though it have been the simplest one within side the gadget. The cappotential of 1 transaction to arise will now no longer have an effect on that of another.

- **Knowledge Check 1**

Fill within side the Blanks

1. When numerous transactions are being finished concurrently and in parallel in a database gadget, the belongings of ________ ensures that every transaction may be treated and finished as though it have been the simplest one within side the gadget.
2. A kingdom alternate may be represented through a transaction. Ideal transactions include the 4 characteristics, known as __________ belongings.

3______________ is the name of a self-contained, reliable, and consistent painting unit that operates against a database using a database control device (or a device similar to it).

- **Outcome-Based Activity 1**

List out all of the acid Properties of Database Transactions.

4.6 Interleaved Executions

During a single-query execution, interleaved execution modifies the unidirectional boundary between the optimization and execution phases and enables plans to adjust in light of the updated estimates. Many users frequently share a DBMS. Transactions from these users can be interleaved to speed up the execution of their queries.

Users do not have to wait for other users' transactions to finish completely before starting their own transaction because queries can be interspersed.

A consistent database could be subjected to committed transactions from certain anomalies related to interleaved execution (DBMS) and become inconsistent as a result. Two operations on the same data object clash if at least one of the operations is a write. Recompiling and executing the remaining piece of the query using the result and result cardinality from a section that may be executed independently is the goal of interleaved execution.

4.7 Schedules

A series of steps taken from one transaction to the next is referred to as a timetable. It keeps the actions of every single transaction in their proper sequence..

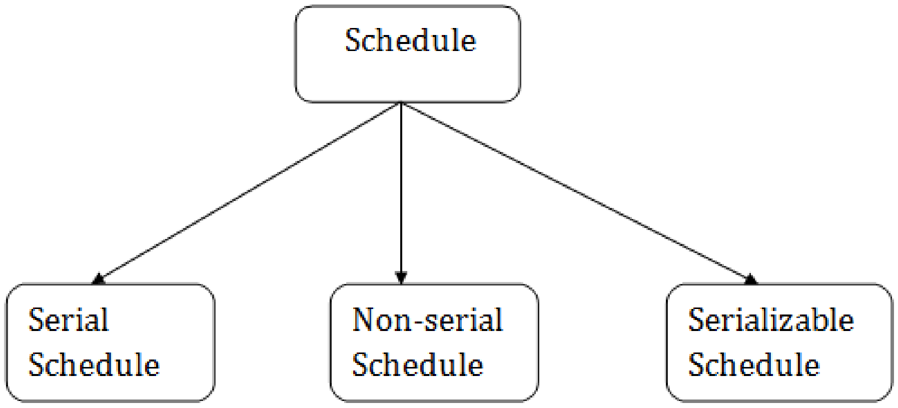


Fig 4.4 Next transaction series

1.Serial Schedule

A serial schedule is a type of timetable that demands that one transaction be completed before starting another. The serial schedule's next transaction is executed after the previous one's cycle is finished.

For instance:

Let T1 and T2, two transactions containing operations, be considered.
In the event that operations are not interleaved, the following two results could occur:
After finishing all of the T1 operations, all of the T2 operations came next. After finishing all of the T1 operations, all of the T2 operations came next.

The accompanying (a) parent, Schedule A, suggests a serial time table wherein T1 and T2 are sequentially ordered. The provided (b) parent suggests Schedule B as a serial time table, with T2 arriving after T1.

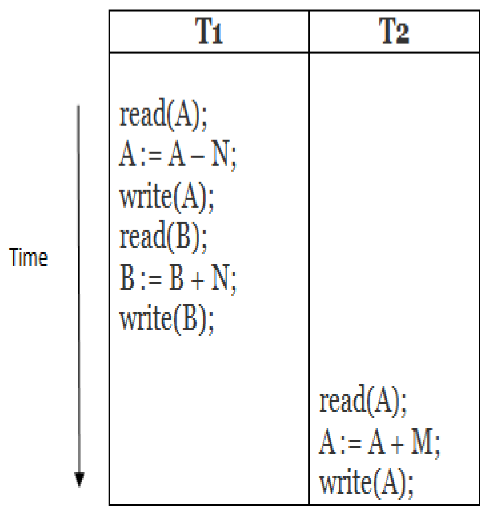
2. Non-serial Schedule

If interleaving operations is allowed, there may be a non-serial time table. It has a massive quantity of feasible orders wherein the machine may want to execute the numerous approaches for the severa transactions. The non-serial schedules C and D may be determined withinside the adjoining figures (c) and (d). There is a lag among the operations.

3. Serializable time table

The serializability of schedules is used to locate non-serial schedules that permit the transaction to execute simultaneously with out interfering with one another. It demonstrates which schedules are correct while the operations of the transaction are accomplished in a non-sequential manner. A non-serial time table may be serialized if the end result of the time table is similar to the end result of its serially accomplished transactions.

(a)



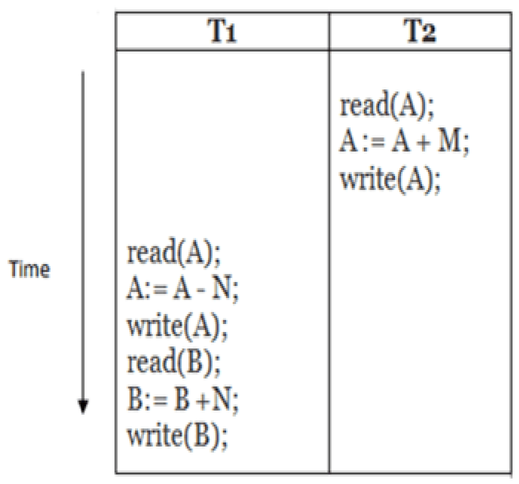
T1	T2

read(A);
A:= A – N;
write(A);
read(B);
B := B + N;
write(B);	

	read(A);
	A:= A + M;
	write(A);

Schedule A

(b)



T1	T2

	read(A);
	A:= A + M;
	write(A);

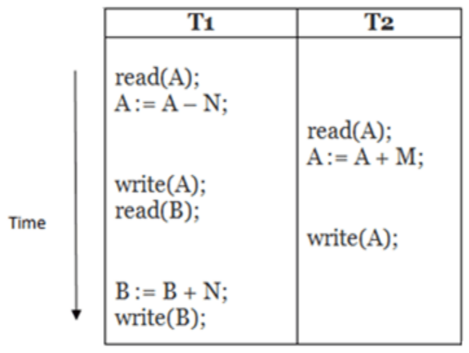
read(A);
A:= A - N;
write(A);
read(B);
B:= B +N;
write(B);	

Schedule B

.

Fig 4.5 Scheduling Of Transaction

(c)



T1	T2

read(A);
A := A – N;	

	read(A);
	A := A + M;

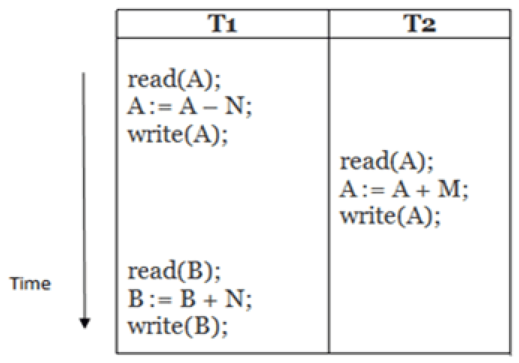
write(A);
read(B);	

	write(A);

B := B + N;
write(B);	

Schedule C

(d)



T1	T2

read(A);
A:= A – N;
write(A);	

	read(A);
	A:= A + M;
	write(A);

read(B);
B := B + N;
write(B);	

Schedule D

Fig 4 6 Scheduling Of Transaction

Schedules A and B are serial schedule. On-serial schedules include Schedule C and Schedule D.

4.8 Serializability

The serializability of a device is a belongings that characterizes how specific approaches use shared information. A device is taken into consideration serializable if there may be no overlap in execution and the end result stays the identical no matter the order wherein the operations are performed.

A device`s serializability defines how numerous approaches engage with shared information.If the final results of the operations is similar to in the event that they had been performed in a non-overlapping, sequential fashion, the device is serializable.

Data may be locked down the use of a database control device (DBMS) to save you different approaches from gaining access to it at the same time as it's far being examine or modified.

Types of serializability

There are kinds of serializability −

1. View serializability

When regarded further to a serial timetable, a time table has view-serializability.

It adheres to the subsequent rules:

- After T1 has completed studying A's preliminary fee, A's preliminary fee is likewise examine through T2.
- T1 reads the fee that turned into written through T2, and T2 then reads the fee that turned into additionally written through T1.
- Following T1's write operation, T2 additionally makes use of the write operation because the very last fee.

2. Conflict serializability

Similar to a few serial execution, it arranges any incompatible operations. If operations proportion the identical information object and one in every of them is a write operation, they're stated to war.

It implies

- Readi(x) readj(x) - non-war examine-examine operation.
- Readi(x) writej(x) - war examine-write operation.
- Writei(x) readj(x) - war write-examine operation.
- Writei(x) writej(x) - war write-write operation.

4.9 Concept of Database Recovery and Backup

The method of producing and maintaining information copies that can be applied to defend corporations in opposition to information loss is called backup and healing. Operational healing is every other time period for this. Database backup regularly includes developing and keeping a duplicate of the database's contents on a backup server which will assure security. Since the database information might be meaningless with out transaction logs, they're additionally stored withinside the backup collectively with the database information.

SixA backup replica of the database is important in case the unique database is corrupted or lost, that is an not likely scenario. This backup lets in you to repair the database to its unique state.

Backup Methods

The numerous database backup strategies include:

Complete Backup - Because a complete replica of the database is made, replete with all the information and transaction information, this approach takes time.

Transaction Log - Only the transaction logs are stored as a backup the use of this approach. To reduce the scale of the backup file, the preceding transaction log information is deleted after a brand new backup report is created.
Differential Backup- It is similar to complete backup in that it continues music of each the transaction information and the information. Regretfully, most effective the information that has modified because the ultimate complete backup is blanketed withinside the backup. Files produced the use of differential backup are consequently smaller.

Recovery Methods

For database recovery, there are basically ways.

Which are:

1. Log-primarily based totally recovery - In log-primarily based totally recovery, every database transaction log is saved in a steady region in order that withinside the case of a gadget failure, the database might also additionally retrieve the facts. All log facts, together with the transaction`s time, contents, etc., ought to be stored previous to the transaction being executed.

2. Shadow paging - When a transaction is finished the use of shadow paging, the facts is robotically preserved for later use. If the gadget crashes withinside the center of a transaction, the database will now no longer replace to mirror the modifications made through the gadget.

• Knowledge Check 2

State True and False for the subsequent Sentences.

1. After a transaction is complete, shadow paging permits the facts to be robotically preserved for later use.

2. Schedules are the technique of making and keeping facts copies that may be applied to guard businesses in opposition to facts loss

3. A collection of steps crafted from one transaction to the subsequent is called a timetable.

Outcome-Based Activity 2

List out all styles of DBMS Schedules.

4.9 Summary

- During question processing, high-degree queries are converted into low-degree expressions. This is a scientific technique that may be used on the record gadget's bodily degree, all through question optimization, and whilst the question is really run to get the favored outcome. It calls for a essential know-how of record control and relational algebra.
- In a database control gadget, a transaction is a unmarried unit of hard work or good judgment that could encompass a couple of operations. Any logical computation accomplished continually is known as a database transaction.
- In the context of transaction processing, the abbreviation ACID stands for atomicity, consistency, isolation, and durability—the 4 essential houses of a transaction. Updates to the facts are handled as though they had been one non-stop activity.
- Interleaved execution permits plans to conform in mild of up to date estimates for the duration of a unmarried-question execution through changing the unidirectional boundary among the optimization and execution phases.
- A collection of steps observed from one transaction to the subsequent is called a timetable. It continues the moves of each unmarried transaction of their right collection.
- The serializability of a gadget describes how extraordinary strategies speak with not unusualplace facts. A gadget is serializable if the end result is similar to if the operations had been completed in a non-overlapping, sequential manner.

4.9 Self-Assessment Questions

1. What is query Processing?
2. Explain the Processing of joins.
3. What is the difference between Materialized and pipelined processing?

4. What is DB transactions?
5. Explain ACID Property.
6. What is Interleaved Executions?
7. What isthe Schedules?
8. What is the concept of database backup and recovery?

4.12 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 5

RELATIONAL DATABASE DESIGN USING PLSQL

Learning Outcomes

- Students will be capable of learning about the PLSQL, Data types, and Language Structure.
- Students will be able to understand about the Controlling of Program Flow and Conditional Statements.
- The ideas of loops and stored procedures will be clear to students.
- Students will be able to understand aboutthe Stored Functions.
- The course will cover handling exceptions and errors for students.
- Students will learn about the Cursors and Triggers.

Structure

5.1 Introduction to PLSQL

Procedural languages like PL/SQL are made especially to support SQL statements in their syntax. PL/SQL software units are assembled and stored inside the database by the Oracle Database server.. Additionally, for maximum efficiency, SQL and PL/SQL execute within the same server process at run-time.

The PL/SQL programming language was created by Oracle Corporation in the latter part of the 1980s as a procedural extension language for SQL and the Oracle relational database.. PL/SQL is a block-structured language.PL/SQL programs consist of logical blocks that can be further subdivided into an infinite number of sub-blocks. The language used by Oracle is known as Pl/SQL, or "Procedural Language extension of SQL."

Oracle database is connected with PL/SQL (since version 7). With each new release of the Oracle database, PL/capabilities SQL's are often increased. Although though PL/SQL and SQL are closely connected, it nonetheless introduces some additional programming constraints that SQL does not provide. Except for string literals and character literals, PL/SQL does not care whether letters are capitalized or lowercase. Lexical units, which are collections of characters, make up a line of PL/SQL text. It falls under the following categories:

1. Delimiters
2. Identifiers
3. Literals
4. Comments

The following are some noteworthy PL/SQL facts:

1. A high-performance, entirely portable transaction processing language is PL/SQL.
2. An integrated, interpreted, and OS-independent programming environment is offered by PL/SQL.
3. Moreover, the SQL*Plus command-line interface allows for the direct invocation of PL/SQL
4. Other programming languages can also make direct database calls.
5. The general grammar of PL/SQL is based on that of the programming languages ADA and Pascal.

6. In addition to Oracle, PL/SQL is supported by IBM DB2 and Times Ten's in-memory database.

- **Features of PL/SQL**

The following characteristics of PL/SQL include

1. SQL and PL/SQL work together quite closely.
2. It provides thorough mistake checking.
3. It provides a variety of data types.
4. It provides a selection of coding structures.
5. Using its functions and procedures, it promotes organized programming.
6. Programming that is object-oriented is supported.
7. It facilitates the creation of server pages and web applications**.**

5.2 PLSQL: Data types, Language structure

1. Data types

• PL/SQL has a big range of predefined data types.A few examples encompass person, Boolean, date, collection, reference, integer, floating point, and massive object (LOB) kinds.Additionally, you could use PL/SQL to outline custom subtypes. In addition to growing SQL queries, PL/SQL data types may be used within side the PL/SQL block, much like in another programming language. The deliver of a data type instructs Oracle on the way to keep and manage any records whilst any PL/SQL code block is executed. A data type identifies the sort of records being used, which include a word (string), a unmarried person, or any other sort of records. The following data types may be utilized in PL/SQL, relying at the sort of records needed:

2. Scalar Types-Basic records kinds called scalar kinds regularly bring a unmarried value, which include a unmarried digit or a string of characters. In the photo above, there are 4 classes for scalar kinds: range kinds, person and string kinds, Boolean kinds, date and time kinds, etc.

1. LOB Types: Large gadgets, which include textual content files, images, and different forms of gadgets which might be normally now no longer saved outdoor of a database, could have their places defined the usage of this datatype.

2. Reference Types: Pointer values, which commonly keep the addresses of different application components, are saved on this form of records.
3. Composite Types:Last however now no longer least, because the call implies, this shape of records is an amalgamation of impartial records that may be treated and processed separately.

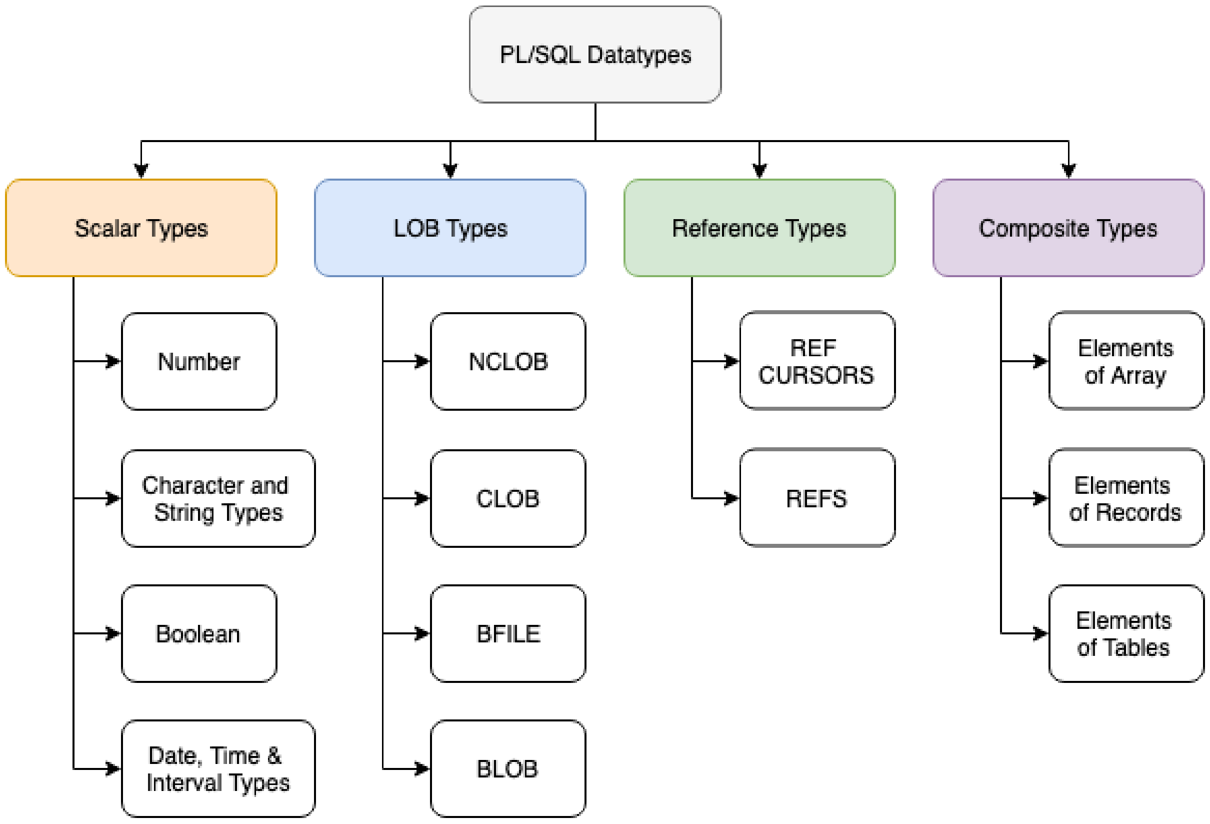


Fig. 5.1 PLSQL Database

1. NUMBER (p, s)

Range: s= -84 to 127, p= 1 to 38

Numerical data is typically stored using this data type. Here, s is scale and p is accuracy.

For instance, Age NUMBER (2), where Age is a variable used to hold two numbers

2.CHAR(size)

Scale: 1–2000 bytes

This datatype stores an alphabetical string of a given length.

Its value is denoted by single quotation marks. Even when the data is not using the memory, it takes up the entire allotted amount of memory.

An illustration might be -rank CHAR (10), where rank is a variable with a maximum character storage of 10.

3.VARCHAR(size)

Range: **1 to 2000 bytes**
Alphanumeric strings of various lengths are stored using this datatype. Single quotations are used to indicate its value. Even when the data is not using the memory, it takes up the entire allotted amount of memory.

Example:
address is a variable that may hold up to 10 bytes of data and has the type VARCHAR (10). It can contain alphanumeric values. Waste occurs in unused areas.

4.VARCHAR2(size)
Range: **1 to 4000 bytes**
Variable-length alphanumeric strings are stored using this datatype. Its value is in single quotation marks. It frees up memory space that isn't being used, hence conserving space.

Example

call is a variable that may keep up to ten bytes of facts in memory, or an alphanumeric cost, the use of the layout VARCHAR2(10). Memory that isn`t getting used is released.

5.Date

Range: 01-Jan-4712 BC to 31-DEC-9999

The data is stored in date layout, DD-MON-YYYY.This datatype's cost is enclosed in unmarried quotes.

Example:

DOB DATE, wherein DOB is a variable that holds the date of beginning in a predetermined layout (for example, "13-FEB-1991")

6. %TYPE

This characteristic saves the cost of a variable whose datatype is unknown however which we want to inherit the datatype of a desk column. It additionally adopts the datatype of the column for which it's far used, on the grounds that its cost is generally received from an present database desk.

Student sno %TYPE; is the call of the desk that became created withinside the database. The variable sno has an unknown datatype, and the cost of %TYPE is used to keep it.

7. BOOLEAN

a. Conditional statements use this datatype.

b. The values are stored logically.

c. Either it's far TRUE, or it's far FALSE.

Example:

Admin is a variable whose cost may be both TRUE or FALSE, relying at the situation being checked. Its kind is BOOLEAN.

- Structure

A block-dependent language is PL/SQL.In different words, the essential constructing factors of a PL/SQL programme (procedures, functions, and nameless blocks) are logical blocks, which could comprise any range of nested sub-blocks. Every logical block commonly refers to a trouble or sub-trouble that desires to be solved. The smallest significant series of code in PL/SQL, like within side the majority of different procedural languages, is called a block. An execution and scoping boundary for variable declarations and exception coping with is furnished through a block of code. In PL/SQL, you may write named blocks, which may be packages, procedures, functions, triggers, or item types, in addition to nameless blocks, which can be code blocks with out names.

A semicolon marks the stop of each PL/SQL query (;). Using BEGIN and END, PL/SQL blocks may be nested interior of different PL/SQL blocks. The essential layout of a PL/SQL block is as follows:

```
DECLARE

BEGIN

EXCEPTION

END;
```

Only one of the 4 additives in a PL/SQL block ought to be mandatory:

1.Header

only utilized for named blocks. The header specifies how to call the named block or programme. It is Optional Part.

2.Section on Declaration

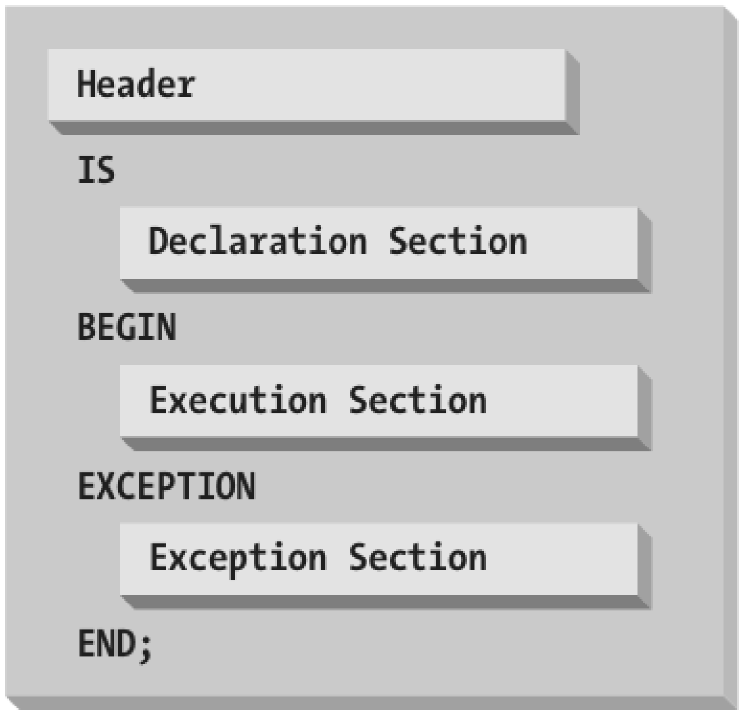
Describes the variables, cursors, and sub-blocks that the execution and exception sections relate to.It is Optional Part.

3.Section of execution

Statements At runtime, the PL/SQL runtime engine will operate.It is Mandatory Part.

4.Section on exceptions

Handles deviations from the standard processing (warnings and error conditions). It is Optional Part.



5.3 Controlling the Program Flow

Sequential control statements, loop statements, and conditional selection statements are the three types of control statements in PL/SQL.

The PL/SQL control statement categories are:

1. Statements for conditional choice, which, primarily based totally at the fee of the enter records, execute a couple of statements. The conditional choice statements are CASE and IF.

2. Statements that repeat the identical commands with one of a kind records values the usage of loops.The essential LOOP, FOR, and WHILE loop statements are the loop statements.

The EXIT assertion actions the point of interest to a loop`s conclusion.

The CONTINUE assertion ends the modern loop new release and palms off manipulate to the subsequent one.

The WHEN clause, that is elective for each EXIT and CONTINUE, lets in you to specify a circumstance.

3. PL/SQL programming does now no longer require sequential manipulate statements. GOTO, which jumps to a selected assertion, and NULL, which does nothing, are the sequential manipulate statements.

4. Four Conditional Statements

diverse records values, the conditional choice statements CASE and IF execute diverse statements. The CASE assertion executes the corresponding assertion primarily based totally on a choice from a listing of circumstances.

These are the paperwork for the CASE assertion:

1. IF THEN
2. IF THEN ELSE
3. IF THEN ELSIF

The CASE assertion executes the suitable assertion through selecting one from a listing of circumstances. These are the paperwork for the CASE assertion:

a. Simple, which compares a unmarried expression to more than a few feasible values.

b. Several situations are assessed through searched, which selects the primary one this is true.

When a separate route of movement is to be taken for every choice, the CASE assertion is appropriate.

1. IF THEN Statement

Depending on a circumstance, the IF THEN assertion executes or skips a sequence of 1 or greater statements.

Here is the shape of the IF THEN assertion:

```
IF circumstance THEN

statements

END IF;
```

The statements execute if the circumstance is true; otherwise, the IF assertion does nothing.

2. IF THEN ELSE Statement

The shape of the IF THEN ELSE assertion is as follows: If the conditional fee is true, the statements execute; otherwise, the else statements execute.

Here is the shape of the IF THEN ELSE assertion:

```
IF circumstance THEN

statements

ELSE

else statements

END IF;
```

3. IF THEN ELSIF Statement

This is the shape of the IF THEN ELSIF announcement:

```
IF condition_1 THEN
statements_1
ELSIF condition_2 THEN
statements_2
[ ELSIF condition_3 THEN
statements_3
]...
[ ELSE
  else statements
]
END IF;
```

The preliminary sentences whose circumstance is proper are done with the aid of using the IF THEN ELSIF clause.

Conditions that also exist aren't assessed. If not one of the situations are met, the IF THEN ELSIF announcement does nothing; otherwise, the else statements, if any, run.

4. Simple CASE Statement

The fundamental CASE announcement consists as follows:

```
CASE selector
WHEN selector_value_1 THEN statements_1
WHEN selector_value_2 THEN statements_2
...
WHEN selector_value_n THEN statements_n
[ ELSE
else statements]
END CASE;]
```

The expression that the selector is (normally a unmarried variable). Every selector price has feasible values: literal and expression. The trustworthy CASE announcement executes the primary set of statements whilst selector price == selector.

Conditions that also exist aren't assessed.

If no selector price equals selector, the CASE announcement increases the predefined exception CASE NOT FOUND and executes else statements if they're present.

five. Searched CASE Statement

The shape of the CASE question is as follows:

```
CASE
WHEN condition_1 THEN statements_1
WHEN condition_2 THEN statements_2
...
WHEN condition_n THEN statements_n
[ ELSE
else_statements]
END CASE;]
```

The preliminary statements that the looking CASE announcement executes whilst the circumstance is proper. The last situations aren't assessed. The CASE announcement increases the predefined exception CASE NOT FOUND if not one of the situations are proper and executes else statements if they're present.

5.5 Loops

There can be instances while you want to execute a code block greater than once. In a function, the primary announcement is accomplished first, accompanied with the aid of using the second, and so forth. Statements are typically accomplished in chronological order. Programming languages have numerous manipulate systems that permit for greater complicated execution paths. In the bulk of programming languages, a loop announcement commonly takes the subsequent form: We can execute a announcement or set of statements numerous instances with the aid of using the usage of a loop announcement.

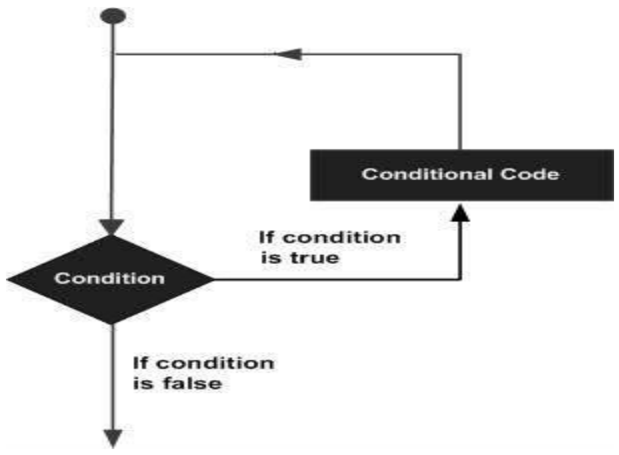


Fig 5.2 Looping in PL/SQL

To meet the looping needs, PL/SQL offers the following types of loops.

1. **PL/SQL Basic LOOP**

   Between the LOOP and END LOOP statements is a sequence of statements that makes up a basic loop structure. The series of statements are carried out on each iteration before the control returns to the loop's beginning.

Syntax:

```
    LOOP
 Sequence of statements;
    END LOOP;
```

A unmarried announcement or a block of statements can be covered on this collection of statements.

The loop have to be damaged the use of both an EXIT or an EXIT WHEN announcement.

2. PL/SQL WHILE LOOP

As lengthy as a exact circumstance is true, a goal announcement is constantly finished the use of a WHILE LOOP announcement withinside the PL/SQL laptop language.

Syntax

```
  WHILE circumstance LOOP
sequence_of_statements
  END LOOP;
```

3. PL/SQL FOR LOOP

The repetition manage shape referred to as a FOR LOOP makes it easy to create a loop that have to run a positive variety of times.

Syntax

```
FOR counter IN initial_value ..final_value LOOP
sequence_of_statements;
END LOOP;
```

The route of manage in a for Loop is as follows:

a. The preliminary step is finished most effective once, on the beginning. At this point, any loop manage variables may be declared and initialized.
b. After that, the circumstance—which includes the beginning and finishing values—is evaluated. If the end result is TRUE, the loop`s frame is carried out. The announcement that follows the for loop assumes manage and skips the frame of the loop if it returns FALSE.
c. counter variable's price is raised or diminished following the execution of the for loop's important frame. The state of affairs is now reevaluated.
d. If it's miles TRUE, the loop is carried out, and the technique is repeated (frame of loop, then increment step, and however circumstance).
e. FOR-LOOP ends while the circumstance is decided to be FALSE.

4. Nested loops in PL/SQL

One loop can be used interior every other in PL/SQL Every different simple loop, while, or for loop can also additionally encompass one or greater loops.

Syntax

```
LOOP
  Sequence of statements1
  LOOP
    Sequence of statements2
  END LOOP;
END LOOP;
```

The Loop Control Statements

Statements used to govern loops divert execution from the standard path. All computerized gadgets produced inside a scope are deleted while execution exits it.

The following manage statements are supported with the aid of using PL/SQL.Putting the manage outdoor of a loop is facilitated with the aid of using labelling the loops.

1.EXIT announcement

Control shifts to the announcement after the Exit announcement completes the loop, which happens after the END LOOP.

2.CONTINUE the announcement

This makes the loop skip the the rest of its frame and right now retest its country earlier than continuing.

3.GOTO announcement

It places the labelled announcement in charge. While the use of the GOTO announcement for your programme isn't recommended.

- Knowledge Check 1

Fill withinside the Blanks

1. As a procedural extension language for SQL and the Oracle relational database, the PL/SQL programming language became created with the aid of using ___________in the overdue 1980s.
2. Sequential manage statements, loop statements, and conditional choice statements are the 3 styles of _________in PL/SQL.
3. Only one of the ___________components in a PL/SQL block have to be mandatory.

- Outcome-Based Activity 1

List out all of the Conditional and Loop statements in PLSQL.

5.6 Stored Procedures

A manner or characteristic is a schema object that logically companies collectively a set of SQL and different PL/SQL laptop language statements to perform a particular task. Often referred to as a "saved manner," a manner is a subroutine this is stored in a database, similar to a subprogram in a conventional programming language.

a. A manner includes a SQL command (s), a listing of parameters, and a name.Procedures and features are created the usage of a user`s schema and saved in a database for next use.

PL/SQL blocks are subprograms inside PL/SQL that can be invoked with many arguments.There are kinds of subprograms to be had in PL/SQL.

b. Functions: These programmers' subroutines normally compute and go back a unmarried value.
c. Procedures: Mostly hired to perform actions, those subprograms do not without delay go back a value.

Creating A Procedure

The CREATE OR REPLACE PROCEDURE declaration generates a manner. The following is the CREATE OR REPLACE PROCEDURE declaration's shortened syntax:

```
CREATE [OR REPLACE] PROCEDURE procedurename
[(parameter name [IN | OUT | IN OUT] kind [, ...])]
 AS
BEGIN
< procedure body >
END manner name;
```

5.7 Stored Functions

Subprograms that may be written and stored withinside the database as database items encompass processes and features. They also can be referenced or invoked interior of different blocks. There are numerous similarities among the PL/SQL Function and PL/SQL Procedure. While a way might also additionally or might not go back a value, a characteristic is needed to accomplish that constantly. The predominant distinction among a technique and a characteristic is this. The PL/SQL characteristic and the PL/SQL manner percentage all different features.

A go back declaration is a demand for the characteristic.

a. The records kind you ought to go back from the characteristic is precise via way of means of the RETURN clause.
b. The executable element is contained withinside the characteristic body.
c. For defining a solo characteristic, the AS key-word is used instead of the IS key-word.

Syntax:

```
1.        CREATE [OR REPLACE] FUNCTION characteristic name [parameters]
2.        [(parameter name [IN | OUT | IN OUT] kind [, ...])]
3.        RETURN returndatatype
4.         AS
```

5. BEGIN
6. < function body >
7. END [function name];

Example

create or update characteristic adder (n1 in number, n2 in number)

go back number

is

n3 number (8);

begin

n3 :=n1+n2;

go back n3;

end;

5.8 Handling Errors and Exceptions

In PL/SQL, an blunders that occurs whilst the programme is jogging is called an exception. Programmers can seize those occurrences the usage of exception blocks in PL/SQL, and the mistake circumstance is treated appropriately. A PL/SQL blunders this is raised at some stage in programme execution, both expressly with the aid of using your programme or accidentally with the aid of using TimesTen, is an exception. An exception may be dealt with with the aid of using propagating it to the caller surroundings or catching it the usage of a handler.

Two classes of exceptions exist:

1. Exceptions described with the aid of using the system
2. Exceptions created with the aid of using the user

Syntax for exception handling:

DECLARE

BEGIN

EXCEPTION

```
   WHEN exception1 THEN
       exception1-handling-statements
   WHEN exception2  THEN
      exception2-handling-statements
   WHEN exception3 THEN
      exception3-handling-statements
   ........
   WHEN others THEN
      exception3-handling-statements
END;
```

5.nine Cursors

When a SQL declaration is processed with the aid of using Oracle, a context place is created in reminiscence.

The pointer to this context place is the cursor. It consists of all of the statistics required to manner the declaration. The context place in PL/SQL is controlled with the aid of using Cursor. The rows of statistics accessed with the aid of using a pick declaration are indexed in a cursor.
A programme is called a cursor whilst it's far used to retrieve and take care of every row again with the aid of using a SQL declaration individually. Cursors are available in varieties:

- Implicit Cursors
- Oracle will generate implicit cursors whilst the SQL declaration is being performed in case you don`t make use of an specific cursor. They are mechanically created whilst DML commands, like INSERT, UPDATE, DELETE, etc., are performed to manner the statements.

To study the country of DML activities, Orcale gives numerous attributes called Implicit cursor's attributes.
%FOUND, %NOTFOUND, %ROWCOUNT, and %ISOPEN are some of them.
- Explicit Cursors

To have extra manipulate over the context place, programmers specify specific cursors.

These cursors want described withinside the PL/SQL block's assertion section.It is constructed on a SELECT operation that produces numerous rows.

Syntax

CURSOR cursor call IS pick declaration;;

Steps

These moves achieved whilst the usage of an specific cursor.

1. Declare the reminiscence cursor's initialization.
2. For reminiscence allocation, pass the cursor.
3. To retrieve statistics, get better the cursor.
4. To launch the allocated reminiscence, near the cursor.

5.10 Triggers

When a particular occasion takes place, the Oracle engine routinely calls the cause. When a sure situation is met, the cause is regularly referred to as from a database. Triggers are saved applications which can be routinely released or finished whilst a particular occasion takes place.

The following occurrences can motive triggers to be written to be finished.

1. The announcement for database manipulation (DML) (DELETE, INSERT, or UPDATE).
2. The announcement is defining a database (DDL) (CREATE, ALTER, or DROP).
3. An operation on a database (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

The desk, view, schema, or database that the occasion is associated with may want to all have triggers described on them.

Advantages of the usage of Triggers

1. Trigger routinely creates a few derived column values.
2. The enforcement of referential integrity
3. Recording activities and preserving music of desk get admission to statistics Auditing
4. Synchronized desk replication
5. The imposition of protection authorizations
6. Avoiding faulty transactions

- Knowledge Check 2

State proper and False for the subsequent Sentences.

1. Often referred to as a program, a method is a subroutine saved in a database that capabilities further to a subprogram in a conventional programming language.
2. When a particular occasion takes place, the Oracle engine routinely calls the cause.
3. In PL/SQL, an mistakess that takes place whilst the programme is going for walks is called an occasion.

- Outcome-Based Activity 2

  Try to Enable, Disable and Drop Triggers withinside the SQL Server.

5.11Summary

- A procedural language referred to as PL/SQL became created with SQL statements already constructed into its syntax. The Oracle Database server builds PL/SQL programme gadgets and shops them withinside the database.
- The use of conditional statements permits you to "test" diverse assumptions after which execute the statements primarily based totally at the results. In PL/SQL, there are sorts of conditional manage statements:
   - IF clauses
   - CASE declarations
- Many predefined datatypes are presented via way of means of PL/SQL.Integer, floating point, character, Boolean, date, collection, reference, and huge object (LOB) sorts are only some examples. You also can create your very own subtypes the usage of PL/SQL.
- A collection of statements inside a PL/SQL code block is repeated the usage of the LOOP announcement.
- Each access into the loop frame is straight away observed via way of means of an assessment of the situation.
- The 4 styles of loop statements presented via way of means of PL/SQL are primary loop, WHILE loop, FOR loop, and cursor FOR loop.
- An utility programme makes use of a cursor, additionally referred to as a named manage structure, to perceive and select out a row of statistics from a end result set. You can use a cursor to examine and execute the question end result set one row at a time in preference to going for walks the question in its whole.

- Indirect Cursors and Direct cursors are styles of Cursors.
- An insert, update, or delete operation on a desk will motive a PL/SQL cause, that is a named database object, to specify and perform a sure set of actions. The CREATE TRIGGER announcement in PL/SQL is used to assemble triggers.

5.12Self-Assessment Questions

1. What is PLSQL?
2. What are PLSQL data types?
3. What are Control Flow Statements in PLSQL?
4. What is Conditional Statements in PLSQL?
5. Explain Loops in PLSQL.
6. What is Stored Procedures and Stored Functions?
7. What is Cursors?
8. What is Triggers?

5.13 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 6

CONCURRENCY CONTROL

Learning Outcomes:

- Students will be capable of learning about the Concurrency Control techniques.
- Students will be capable to understand the Locking and management of locks.
- Students will be capable to understand the concepts of 2PL.
- Students will be a capable to ble to understand about the Locking Techniques deadlocks.
- Students will be ab capable to le to learn about Optimistic Concurrency Control.
- Students willlearn about the Comparison of Concurrency control methods.

Structure

6.1 Concurrency Control Techniques

In the database control device, an concept of concurrency manipulate is largely categorised as Transaction (Database Management System). This are DBMS capabilities which lets in us to manipulate or greater concurrent techniques so that every one of them can run with none war with every other, that is a not unusualplace characteristic in multi-person systems. The idea of Concurrency manipulate is a perception discovered in database control systems. The idea of concurrency manipulate comes beneathneath the Transaction withinside the database control device (DBMS).

It is a DBMS method which lets in us to govern concurrent techniques in the sort of way that they run with out war with every other, that is not unusualplace in multi-person systems. Concurrency is basically the execution of many transactions at as soon as.

It is critical to enhance time complexity and efficiency. Inconsistencies broaden whilst many transactions seeking to get entry to the identical information. To hold information consistency, concurrency manipulate is critical. For example, if we make use of ATM machines with out concurrency, severa human beings can not withdraw cash on the identical time at exceptional locations. This is wherein we require cooperation.

Advantages

Following are the advantages of concurrency manipulate:

- There could be no or much less waiting.
- Reaction time could be decreased and slowed down.
- Uses of sources will rise.
- The efficiency & overall performance of the device are improved.

It is manner in a DBMS which facilitates us lots for the green control of the 2 simultaneous concurrent techniques to execute with out conflicting among one another; those conflicts might also additionally arise in all multi-person systems. All Concurrency can virtually be taken into consideration to be executing numerous a couple of transactions on the identical time. So that is required that we must growth time efficiency.

1. Locking

Lock guarantees that a modern transaction has sole get entry to to positive information objects. It to start with obtains a lock to benefit get entry to to the information items, and as soon as the Transaction is over, it releases the lock.

Various Locks

The many lock sorts are as follows:

- Shared Lock [Data item values can be read only in Transaction]
- Exclusive Lock [Used in both read as well as write values of data item]

2. Time Stamping

A time stamp is the unique identifier that DBMS created, and it indicates specifically the approximate graduation time of the transaction. Any transaction we carry out shops the Transaction`s starting time and shows a specific time. A device clock or logical counter may be used to generate this. This can start whenever a transaction begins. Here, after a sparkling timestamp has been assigned, the logical counter is increased.

The timestamp of information object's may be one of the sorts:

i. W-timestamp(X): Indicates the maximum current time at which the information object X become written into.

ii. R-timestamp(X): The maximum current time the information object X has been examine from is proven with the aid of using this.
Each time whilst any a hit examine/write operation is accomplished at the information object X, those 2 timestamps are updated.

3. **Optimistic**

   It is always predicated on an idea that since conflicts are uncommon, so allowing transactions to move in forward without adding any delay to assure that serializability is rather more effective**.**

4. **Multiversion Concurrency Control:**

   To increase concurrency to a great extent, multiversion schemes retain previous iterations of data items. Each successful write generates an all new version of the data item written in multiversion two-phase locking. The versions are identified by timestamps. Choose the proper version of X when a read(X) operation is performed based on the timestamp of the Transaction.

5. **Validation Concurrency Control**

   On the premise that most database operations don't clash, the optimistic method is based. Both locking and time stamping approaches are unnecessary with the optimistic approach. A transaction is instead carried out without limitations up until it is committed. Each Transaction passes through two or three read, validate, and write phases using an optimistic method.

6.2 Locking and management of locks

Any transaction on this form of protocol should first acquire the right lock at the records earlier than it could examine or write it.In DBMS, lock-primarily based totally protocols prevents transactions from studying and/or writing records till it has acquired the right lock.

By isolateing or locking a selected transaction to at least one user, lock-primarily based totally protocols continually serve to get rid of all of the concurrency trouble in DBMS for all simultaneous transactions.

Two awesome varieties of locks exist:

1. Shared lock:

A examine-handiest lock is extraordinary call for it.The records object may be examine with the aid of using the Transaction in a shared lock. It may be shared due to the fact whilst a transaction keeps a lock, it's far not able to replace the records at the records object.Only energetic shared locks are permitted. A shared or examine lock continually restricts another manner to are seeking for a write lock on the desired segment of the record.A shared or examine lock prevents another manner from searching for a write lock on a record`s described segment. Other manner, on the alternative hand, can continually request a examine lock.

Read integrity is continually supported with the assist of shared locks. Shared locks make sure that a document isn't being changed whilst a examine-handiest request is being made.Shared locks may be extensively utilized to save you a document from being up to date among the time it's far examine and the subsequent syncpoint.A shared lock on a useful resource may be held with the aid of using a couple of responsibilities at once.Although numerous methods can very own shared locks, there are unique situations wherein responsibilities should watch for a lock:

- If extraordinary assignment currently holds the unique lock at the useful resource, then a request for the shared lock have to be delayed.
- If any other responsibilities currently very own shared lock in this precise useful resource, a request for an unique lock should wait.
- If different gift assignment is awaiting an all unique lock for a useful resource whcih already has a shared lock, than a brand new request for a shared lock should be delayed.

2. Exclusive lock

The records object may be examine and written with the aid of using the Transaction whilst withinside the unique lock. This lock is unique, and extraordinary transactions can't edit the identical records on the identical time whilst it's far locked.An unique or write lock offers a manner unique get admission to to the described part of the record for writing.No different manner can lock that segment of the record whilst a write lock is in place.Exclusive locks may be activated or deactivated.

Exclusive locks protect each recoverable and non-recoverable record useful resource modifications.Just one Transaction can very own them at a time.If any other assignment currently has an unique lock or a shared lock in opposition to the asked useful resource, any transaction requiring an unique lock should wait.

6.three 2PL Locking Protocol
Each Transaction includes locking and unlocking the records object twice.

Growth Phase: During this stage, all of the locks are dispersed in specific. Locks are not released until all alterations to these data items have been committed, at which point the second phase—especially the shrinking phase—begins.

Phase of shrinkage: No locks are issued during this time, and before locks are released, any modifications to data items must be recorded (stored).

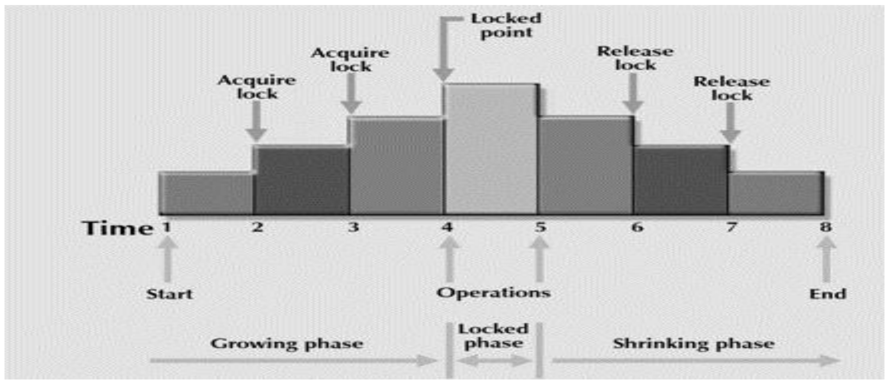


Fig. 6.1 Transaction Growth Phase

When a transaction in the growth phase acquires every lock it could need, it hits its peak. We call this place LOCK POINT. Once the transaction reaches the lock point, it experiences a shrinking phase.
Categories
Two-phase locking comes in two varieties.
1. The rigorous process for two-phase locking
A transaction can release a shared lock following the lock point, but not an exclusive lock until the transaction commits. With this method, fewer cascades occur in the schedule.

a cascading timetable In a cascading schedule, a transaction in this schedule depends on another transaction. As a result, if one rolls back, the other must too.

2. The rigorous two-phase locking protocol

Every lock, whether shared or exclusive, cannot be released by a transaction until it commits. Serializability is guaranteed by the 2PL protocol, but deadlock cannot be prevented.

6.four Locking Techniques

Four unique locking protocols are available:

1. Simplistic lock protocol

The simplest manner to fasten the statistics whilst a transaction is to do it this manner. All transactions can attain the lock at the statistics earlier than deleting, inserting, or updating way to easy lock-primarily based totally protocols. After the Transaction is finished, it's going to release the specific statistics item.

2. Pre-claiming Lock Protocol

Pre-claiming Lock

In order to discover all of the statistics items on which they require locks, protocols verify the Transaction. It asks the DBMS for a lock on every of these statistics items earlier than beginning the Transaction`s execution. TheTransaction can begin beneathneath this protocol if all of the locks are granted. All locks are launched as soon as the Transaction is finished. This protocol permits the Transaction to roll lower back if all of the locks aren't supplied and waits till they may be all granted.

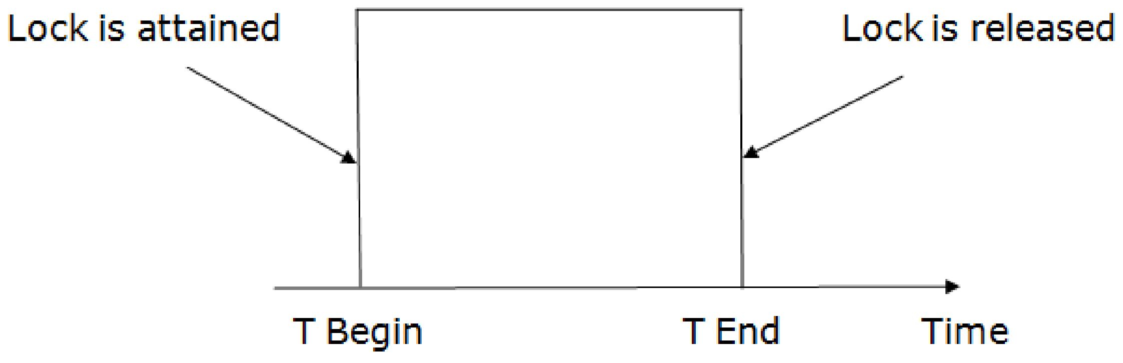


Fig. 6.3 **Pre-claiming Lock Protocol**

3. **Two-segment locking (2PL)**

The execution segment of the Transaction is cut up into the 3 sections through the 2 segment locking protocol. When the unique Transaction`s execution starts offevolved withinside the first step, it asks permission to get the lock it needs. The deal purchases all the locks withinside the 2nd segment. As quickly because the Transaction loses its first lock, the 0.33 segment starts offevolved. The Transaction can't call for any in addition locks at some stage in the 0.33 segment. Just the obtained locks are released**.**

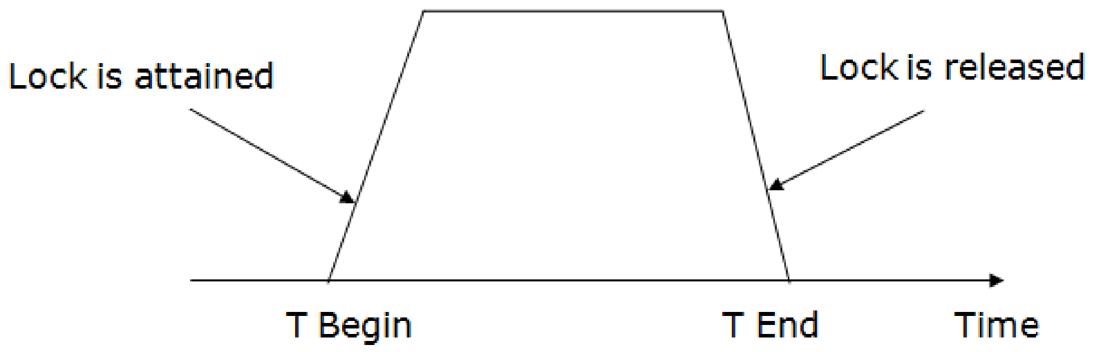


Fig. 6.4 Two-segment locking in transaction

The 2PL technique contains stages:

- Growing section: A new lock at the statistics object can be obtained via way of means of the Transaction at some point of the developing section, however none can be released.
- The section of shrinkage: During the section of shrinkage, any locks already held via way of means of the Transaction can be released, however no extra locks can be purchased.

4. Strict Two-section locking (Strict-2PL)

Strict-2PL`s preliminary section resembles 2PL in sure ways. After acquiring all of the locks withinside the first step, the Transaction maintains to run smoothly. Only the truth that Strict-

2PL does now no longer launch a lock after utilising it distinguishes 2PL from 2PL strictly. Strict-2PL releases every lock personally after watching for the whole Transaction to commit. There isn't anyt any shrinking step of lock launch withinside the strict-2PL technique. As against 2PL, it lacks cascading abort

4. Strict Two-section locking (Strict-2PL)

Strict-2PL`s preliminary section resembles 2PL in sure ways. After acquiring all of the locks withinside the first step, the Transaction maintains to run smoothly. Only the truth that Strict-2PL does now no longer launch a lock after utilising it distinguishes 2PL from 2PL strictly. Strict-2PL releases every lock personally after watching for the whole Transaction to commit. There isn't anyt any shrinking step of lock launch withinside the strict-2PL technique. As against 2PL, it lacks cascading abort

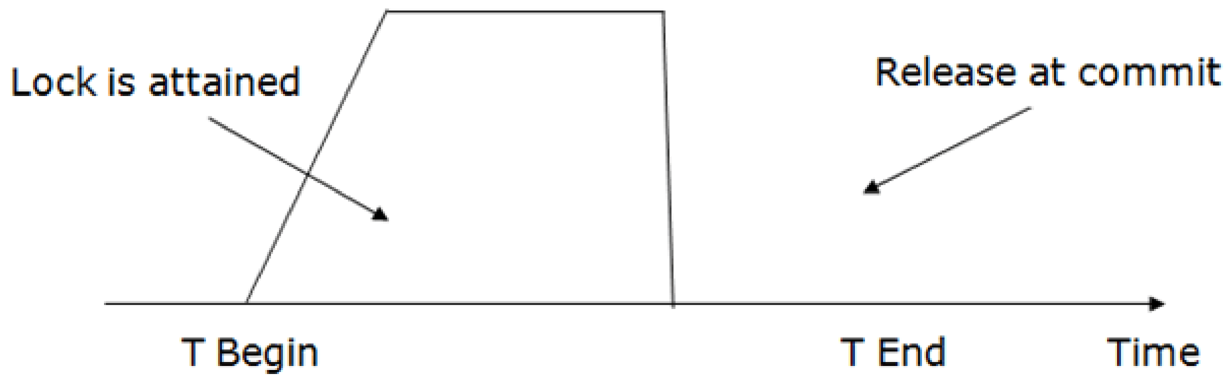


Fig 6.5 Strict Two-section locking in transaction

- Knowledge Check 1

Fill withinside the Blanks for the Following Sentences.

1. It is DBMS method which lets in to manipulate ________concurrent approaches so they run with none war with each other, which in flip is not unusualplace in multi-person systems.
2. By separating or locking a selected transaction to a_________, lock-primarily based totally protocols can serve to get rid of all of the concurrency troubles in DBMS for simultaneous transactions.
3. During the ___________ any locks already held through the Transaction can be released, however no extra locks can be purchased.

- Outcome-Based Activity 1

List out all of the varieties of Locking Techniques in DBMS.

6. Five Deadlocks

A impasse specifically in a database can arise while most of the transactions are anticipating each other to launch locks. For instance, Transaction A might also additionally have a lock on a few rows withinside the Accounts desk and require the updating of positive rows withinside the Orders desk to be finished. When or extra transactions are ready in parallel with no end in sight for each other to launch locks, then this scenario is being known as as a impasse.
One of the maximum well-known and dreaded problems with DBMS is the impasse, which usually prevents duties from ever being finished and specifically maintains them in an limitless ready nation indefinitely.

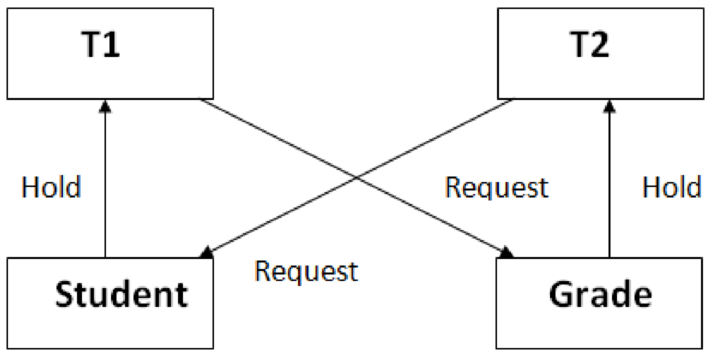


Fig. 6.6 Deadlock in DBMS

Let's say Transaction T1 has to hold the lock on some rows in the student's table while updating some rows in the grade table. Transaction T2 is simultaneously holding locks on those rows in the Grade database and updating those rows in the Student table that Transaction T1 has locked. The main problem now shows up.

At present, Transaction T1 is awaiting the release of its lock from Transaction T2, while Transaction T2 is reciprocating for Transaction T1. Every action comes to a halt and stays still. The DBMS will remain in a standstill unless it detects the impasse and ends one of the transactions.

6.5.1 Deadlock Avoidance

Avoiding a database rather than terminating it or restating it is preferable when it is in a deadlock state. Time and resources are being wasted on this.A system for deadlock avoidance is used to anticipate any deadlock situations.

When a stalemate occurs, a technique called "wait for graph" is utilized, but this technique is only appropriate for smaller databases. Deadlock prevention techniques can be used to larger databases.

6.5.2 Deadlock Detection

A transaction that waits endlessly for a lock should be able to be identified by a database management system (DBMS) as being in a deadlock or not. The lock manager retains a Wait for the graph in order to determine which deadlock cycle exists in the database.

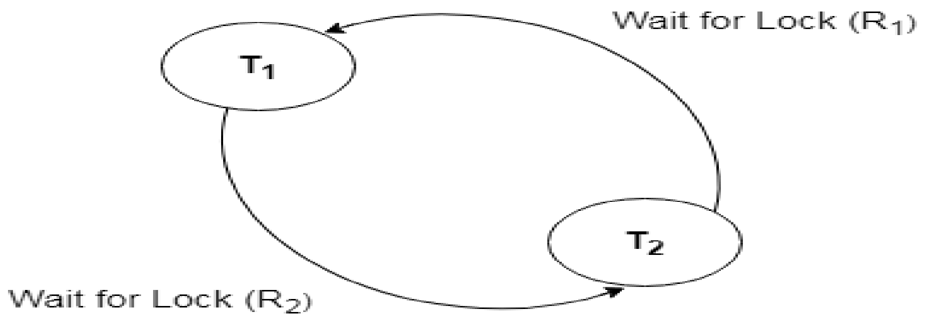


FIG 6.7 Deadlock Prevention

A huge database can use a deadlock prevention approach. The occurrence of deadlock can be avoided if resources are allotted so that it never happens. The database management system examines the transactional operations to determine whether a deadlock situation is likely to arise. If they do, the DBMS never permitted the Transaction to be carried out.

1.Wait-Die scheme

With this system, the DBMS truely compares the timestamps of the 2 transactions if a transaction needs a useful resource this is presently being held with a conflicting lock through any other transaction. It allows the older Transaction to put off the execution till the useful resource is accessible.

2. Wound wait scheme

In a wound wait strategy, the more youthful Transaction is pressured to kill the Transaction and launch the useful resource if the elder Transaction asks a useful resource that it's far holding. The more youthful Transaction is reopened with the equal timestamp after the little pause. If the more youthful Transaction requests a useful resource that the elder Transaction has retained, the more youthful Transaction is told to attend till the older Transaction releases the asked useful resource.

6.6 Optimistic Concurrency Control

Optimistic Concurrency Control, generally called constructive locking, is a concurrency manage method especially utilized in transactional structures which includes software program transactional reminiscence in addition to relational database control structures.OCC makes the idea that numerous transactions can generally end one after any other with out interfering.

Phases

The three steps of optimistic concurrency control are described here.

1.Read phase

Reading and storing various data elements in temporary variables (local copies). These variables undergo all actions without causing the database to be updated.

2. Validation Phase

To verify that serializability will not be validated if the transaction updates are actually made to the database, all concurrent data items are examined. TheTransaction rolls back if the value is altered. The write-sets and read-sets are kept up to date, and the transaction timestamps are used. The following conditions must exist in order to verify that Transaction A does not obstruct transaction B.

- ✓ Before TransA begins the read phase, TransB has finished its write phase.
- ✓ TransB's write phase is finished before TransA's write phase begins, and there are no shared items between the read and write sets of the two transactions.
- ✓ TransB finishes reading before TransA finishes reading, and the read set and write a set of TransA have no items in common with TransB's write set.

3.Write phase

Upon successful validation, the Transaction updates the database. In such a case, transactions are stopped and restarted while updates are discarded. As it doesn't employ any locks, it is deadlock-free; however, data item starvation issues could happen.

6.7 Comparison of Concurrency control methods

The coordination of a transaction`s simultaneous execution in a multi-consumer database gadget is referred to as concurrency manage. The number one purpose of concurrency manage is to guarantee the serializability of the Transaction in a multi-consumer database gadget. The vital facts portions can handiest be accessed in a single, together special manner. That is, a transaction can alter a facts object whilst it's miles being accessed through any other transaction. The maximum famous manner to obtain this criterion is to limition get right of entry to to a facts object to transactions which can be presently in ownership of a lock on it.

One-section or -section locking techniques may be used with locking-primarily based totally concurrency manage systems.

- One-section Locking Protocol
- Two-section Locking Protocol
- Distributed Two-section Locking Algorithm

- Distributed Timestamp Concurrency Control
- Conflict Graphs
- Distributed Optimistic Concurrency Control Algorithm.

There are Various techniques of concurrency manage

1) Binary Locking

There are numerous methods to fasten a facts object:

Shared mode (S): When any transaction Ti has zero shared mode lock on any facts object Q, Ti is handiest capable of study Q; writing to Q isn't always possible.

Exclusive mode (X): Every Transaction in special mode (X) can handiest study and write to Q.If a transaction Ti has a lock on a positive facts object Q this is marked exclusively (X).

The request is despatched through the Transaction to the concurrency manage supervisor. It can handiest flow ahead if the concurring manage supervisor offers the Transaction the lock.

2) Locked-primarily based totally protocol

In a locked-primarily based totally protocol, the middle guiding principle is that a lock need to be acquired earlier than gaining access to a facts object, and the facts object must be deleted at once after use.We are capable of save you a collision as a result. Several locks are used on this Transaction. Exclusive lock and shared lock.

3) Rigorous 2-phase locking

With strict two-phase locking, shrinkage phases are eliminated from the system, making it impossible for a dirty read transaction to acquire locks but release any locks. Although this protocol guarantees conflict, view serializability, recoverability, and cascadelessness, it might experience deadlock.

4) Strict 2-section locking

Strict two-section locking, that's an boost above rigorous two-section locking, lets in the liberating of a shared lock for the duration of the shrinking section.

5) Conservative 2-section locking

With conservation two-section locking, the device`s increase section is removed, and every Transaction ought to first collect all of the locks earlier than wearing out any reads or writes.

It ensures impasse independence, war serializability, view serializability, and war serializability however suffers from recoverability and cascades.

6) Timestamping protocol

With this strategy, we pick to reserve every Transaction earlier than it's far normal into the device. Each Transaction is given a time stamp, Ts(Ti), that's absolutely the present day time at the device on the time the Transaction is normal. As lengthy because the Transaction isn't always but withinside the device, this time stamp may not change. Enter every piece of records. Use the related two-time stamps in Q. Q's timestamp may be examine and written.

- **Knowledge Check 2**

State True and fake for the subsequent Sentences.

1. A impasse in a database takes place whilst one transactions are awaiting every different to launch locks.
2. If a transaction has acquired a shared lock on a records object Q, it may best execute examine operations at the object on its own, however if it has acquired an one of a kind lock, it may carry out each examine and write operations.
3. In a wait-die strategy, the more youthful Transaction is forced to kill the Transaction and launch the aid if the elder Transaction asks a aid that it's far holding.

Outcome-Based Activity 2

Describe all of the kinds of Deadlock Prevention Techniques in DBMS.

6.8 Summary

- In a shared database, the concurrent control technique is a way to control how many transactions are executed at once. The read-write operation of transactions causes conflicts, which are resolved by enforcing the isolation of various transactions and maintaining database consistency.
  - A data variable that is linked to just one data item is referred to as a lock. This lock denotes that data item operations are permitted. Locks fall into one of two categories: Shared and individual locks.
  - Database management systems use the two-phase locking (2PL) protocol, commonly referred to as basic 2PL, to lock data from running concurrent transactions. For instance, one user could need to read data from a record while another tries to modify or update that data at the same time.
  - A deadlock happens when multiple transactions are waiting for their locks in a database to be released. For example, Transaction A may need to change some rows in the Accounts table and have a lock on some entries in the Orders table in order to finish. Optimal concurrency control (OCC), also known as optimistic locking, is a concurrency control technique used in transactional systems such as software transactional memory and relational database management systems.
  - Several transactions can typically complete without interfering with one another, according to OCC.

6.9 Self-Assessment Questions

1. What is concurrency control Techniques?
2. What is Locking in DBMS?
3. Explain 2PL.
4. What are the different types of Locking Techniques?
5. What is deadlocks?
6. What is Optimistic Concurrency Control Techniques?
7. Explain the Management of Locks.
8. What are the different types of Locks?

6.10 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 7

OTHER DATABASES

Learning Outcomes:

- The ability to learn about distributed and parallel databases will be available to students.
- Learners will be able to comprehend what object-based databases are.
- The ideas of XML databases will be understandable to students.
- Learners will be able to comprehend what a NoSQL database is.
- Big Data databases will be taught to the students.

Structure

7.1 Introduction to Parallel and Distributed Databases

1.Parallel Database

A parallel database management system (DBMS) makes use of many processors and is developed to execute function concurrently wherever possible. A parallel database management system (DBMS) links many smaller computers to achieve the similar throughput as a single, large system. Queries are processed concurrently with data loading. Centralized, client-server database systems are not robust enough to support applications that need to handle data quickly. Partial database systems are very beneficial for online transaction processing and decision support applications. A large task is divided into smaller ones in parallel processing, and each smaller work is carried out simultaneously on various nodes. As a result, a bigger task can be finished more rapidly.

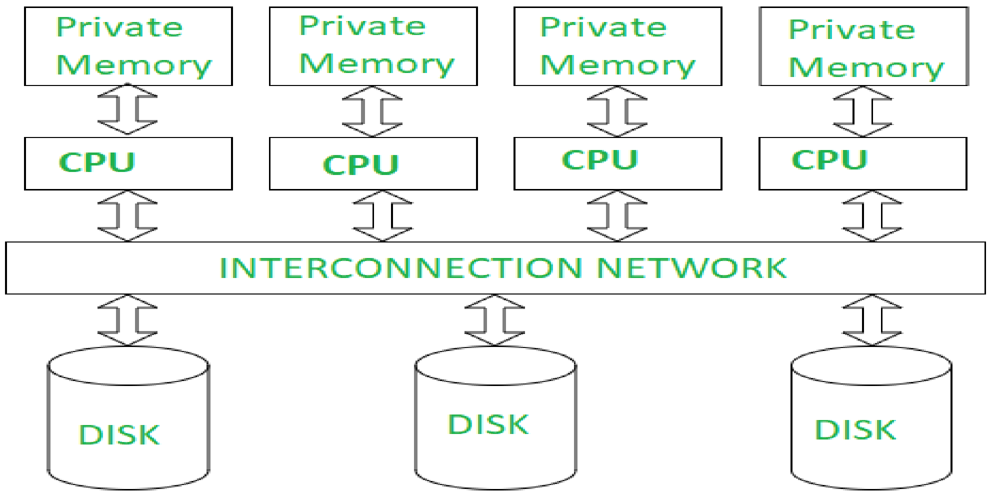


Fig. 7.1 Parallel Database

Features of Parallel Database

- There is CPU parallel processing.
- It breaks down large jobs into numerous smaller activities, which enhances performance.
- It Finishes tasks very rapidly.

Architectural Models

For parallel machines, there are numerous architectural paradigms.

The following are the most crucial ones:

- **Shared-memory multiple CPU**-In this case, the computer has numerous CPUs that are all operating at once and connected to a network for communication. These CPUs share a single main memory and a single disc storage array.
- **Shared disk architecture-**In this design, all nodes share mass storage, but each has their own primary memory. Each node actually has a number of processors.
- **Shared nothing architecture**-Each node in a shared-nothing architecture has its own main memory and mass storage.

Distributed Databases

A distributed database is a shared, specially "connected collection of data that is physically scattered throughout a computer network at several locations".

The software that makes distributed data accessible to users and enables the management of distributed databases is known as a distributed database management system (DBMS).

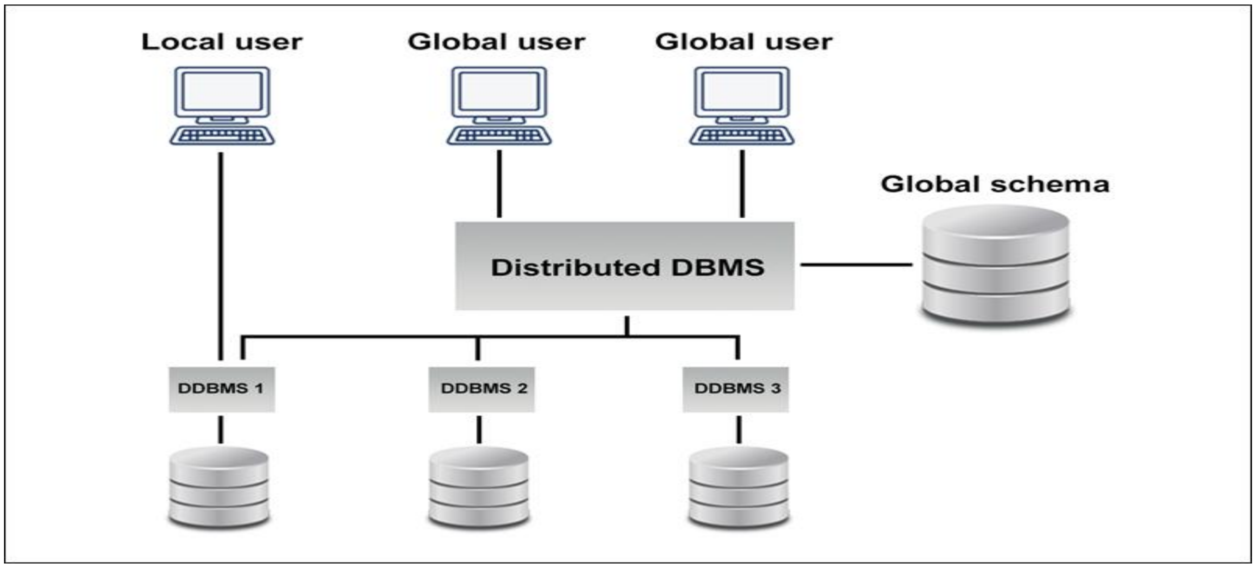


Fig. 7.2 Distributed Databases

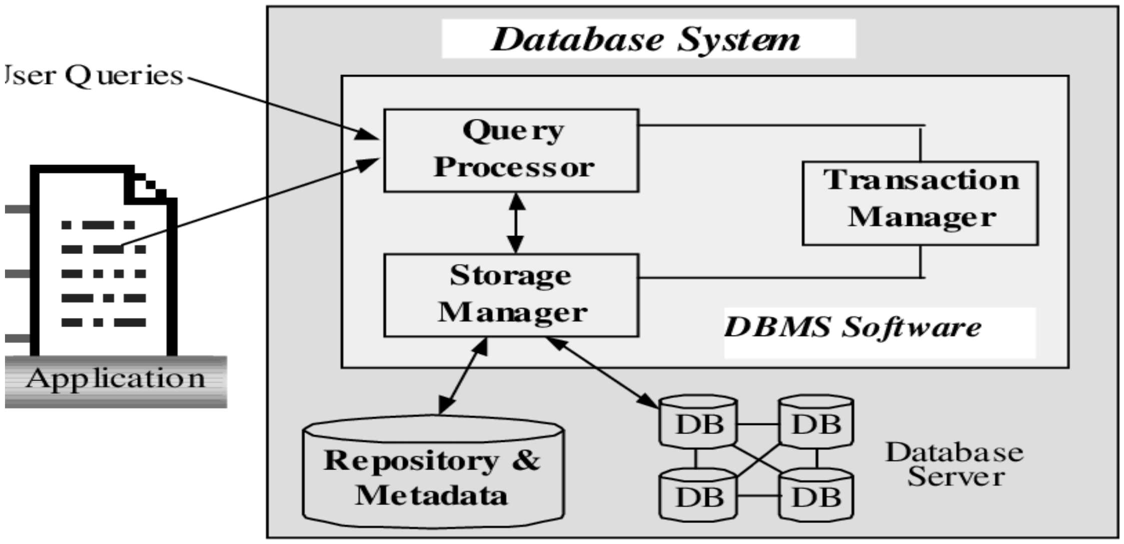


Fig 7.3 Features of Parallel Database

Features of Distributed Databases

- A collection of information that is communicated logically.
- The information is divided into numerous pieces.
- There could be fragment replication.
- There is a communication network connecting the locations.
- The phrase "modular development" refers to the idea that adding nodes to the existing network will allow us to expand the functionality of the system we are now using to additional locations without interfering with it.
- Improves Reliability The phrase "increases reliability" means that if one node on a network fails, the system may continue running by dividing the work among the other nodes in the network.
- Increases Performance– A huge database is divided into smaller databases that are scattered across many places and are easier to manage with greater performance. We all know that a little database is easier to handle than a large database.
- Increased Availability-As data may be accessed from numerous different network nodes, the failure of one node won't have an impact on the availability of the data.
- Faster Reaction – When data is accessible locally, data retrieval becomes more effective.

The Primary distinction among the parallel and dispensed databases is that the previous is tightly coupled after which later loosely coupled.

7.2 Introduction to Object-Based Databases

An item-orientated database (OOD) is a form of database device which can control complicated records items, or items that replicate the ones utilized in item-orientated programming languages. In item-orientated programming (OOP), each aspect is an item. The boundaries imposed via way of means of the relational paradigm have become obtrusive as database structures observed huge use in fields like geographic records structures and computer-aided design. As a remedy, item-primarily based totally databases had been created, allowing the coping with of complicated records types. In item-primarily based totally databases, each entity is seemed as an item and is saved in a table. Inverse reference principle is used to maintain relationships among items and classify associated gadgets into training and subclasses.

Object-Based Data Models

More accurately, the records version is a set of conceptual gear for representing records, records relationships, records semantics, and records constraints. It bureaucracy the premise for the shape of a database.These days, item-orientated programming is the maximum extensively used technique for growing software. Consequently, the ER version turned into extended to encompass notions like encapsulation, techniques, and item identity, ensuing withinside the introduction of an item-orientated records version. The essential principles of item-orientated programming, which include as inheritance, item recognition, encapsulation (records hiding), and techniques for offering items with an interface, had been used to records modeling. The item-orientated records paradigm additionally helps a extra great kind device that consists of series and dependent types. The majority of database vendors presently assist item-relational records fashions, which can be records fashions that blend the traits of relational and item-orientated records fashions. It extends the conventional relational paradigm with numerous traits, consisting of item orientation, dependent and series types, and others.

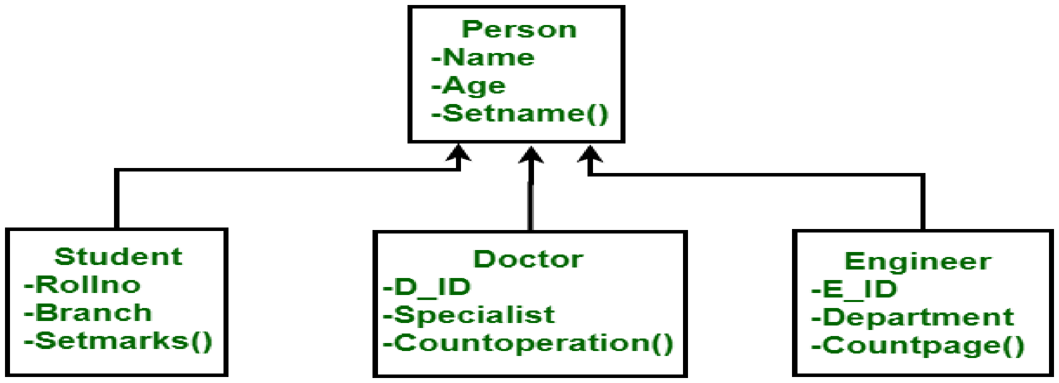


Fig. 7.4 Features of Distributed Databases

Advantages of Object-Based Data Models

• Data that can be easily saved and quickly recovered; data that is more varied and complicated than MySQL data types.

• Object-oriented programming languages may interact with it seamlessly.

• Complex real-world issues are easy to model.

• Extension is made possible by custom data types.

Disadvantages of Object-Based Data Models

- It is not used as frequently as relational databases.
- There's no global data model. There are no theoretical norms or grounds.
- It contradicts the ideas.
- Performance issues result from high complexity.
- There are no adequate access controls or security measures in place for the objects.

7.3 XML Databases

Massive quantities of records in XML layout are saved in XML databases, a kind of records staying power software program gadget. It affords a stable vicinity for XML report storage. You can export and serialize your saved records into the layout of your preference the usage of XQuery. XML databases and report-orientated databases are regularly utilized in tandem. XML databases are used to shop tremendous volumes of records in XML layout. The want for a secure vicinity to keep the XML files arises from the truth that increasingly industries are the usage of them. XQuery may be used to question the database, serialize the records, and export it in any layout required.

Types

XML databases are available extraordinary varieties.

• XML-enabled database

• An XML-enabled database is an extension supplied for the conversion of an XML report. Data is saved in tables which include rows and columns in a relational database which include this one. A series of records, which might be made of fields, are blanketed withinside the tables. A relational database and an XML-enabled database feature in precisely the equal way. It seems like an extension this is to be had to transform XML files. Data is saved in tables with rows and columns on this database.

• Native XML database (NXD)

• Native XML databases keep huge volumes of records. Table shape isn't always the inspiration of local XML databases; instead, boxes are. XPath expressions are used to get right of entry to records. Native XML databases are desired over XML-enabled databases because of their extra capability for storing, handling, and querying XML content. Native XML databases are queried the usage of route expressions. The advantage of an XML-local database over one this is XML-capable. Storing, querying, and handling an XML report is a long way extra effective than an XML-enabled database.

Example

```
<? xml version="1.0"?>
<contact-info>
6
19
34
35
42<contact1>
<name>Arun Sharma </name>
<company>SSMT.org</company>
<phone>9588306809</phone>
```

```
</contact1>
<contact2>
<name>Manish Gupta </name>
<company>SSMT.org</company>
<phone>09990449645</phone>
</contact2>
</contact-info>
```

Knowledge Check 1

Fill withinside the Blanks for the subsequent Sentences.

1. A records staying power software program gadget known as an ________is used to shop the tremendous quantity of records in XML layout.
2. 2. A _______________ is a kind of database gadget which could control complicated records gadgets, or gadgets that mimic the ones observed in object-orientated programming languages.
3. A shared, logically linked series of records this is bodily dispersed throughout a laptop community at diverse places is thought as_______________.

Outcome-Based Activity 1

Illustrate structure of Parallel and disbursed databases with Proper diagram.

7.4 NoSQL Database

NoSQL databases use files rather than relational tables to save information. We consequently categorize them as "now no longer most effective SQL" and similarly separate them primarily based totally on quite a number adaptable information systems. Key-fee stores, graph databases, extensive-column databases, and natural file databases are examples of NoSQL database types.

NoSQL is a sort of database control device (DBMS) that could cope with and save huge quantities of unstructured and semi-based information. Unlike conventional relational databases, which hire tables with pre-described schemas to save information, NoSQL databases use bendy information fashions that could adapt to modifications in information systems and scale horizontally to deal with developing quantities of information. The time period "NoSQL" at the beginning denoted "non-SQL" or "non-relational" databases; but, as NoSQL databases have accelerated to embody a extensive variety of different database architectures and information formats, the time period has come to mean "now no longer simply SQL."SimpleDB, MongoDB, and DynamoDB are the various pinnacle NoSQL database examples."NoSQL database" refers to a non-SQL or non-relational database.It offers an opportunity to the tabular members of the family paradigm utilized by relational databases for information garage and retrieval. NoSQL databases do now no longer save information in tables. Large quantities of information and real-time net programs are commonly hosted there.

History

Flat File Systems are in use on the begin of the 1970s.The predominant problems with flat documents are that every company makes use of its personal flat documents and that there aren't anyt any standards. Data became stored in flat documents. Because there may be no standardized technique for storing information, it's far quite difficult to keep information in documents and retrieve information from documents.

After that, E.F. Codd evolved the relational database, which supplied a method to the problem of the dearth of a not unusualplace technique for storing information. However, relational databases later evolved the problem of being not able to deal with huge quantities of information. As a result, a call for for a database that would cope with all issues arose, and the NoSQL database became created.

E.F.Codd`s relational database became created so as to triumph over the dearth of a standardized mechanism for information storing. However, relational databases later evolved the problem of being not able to deal with huge quantities of information. As a result, a call for for a database that would cope with all issues arose, and the NoSQL database became created.

When to hire NoSQL:

- When storing and retrieving a huge quantity of information is required.
- The connections among the information you save aren't genuinely significant.
- The information is unstructured and evolves over time.
- At the database level, guide for constraints and joins isn't always necessary.

- You should often scale the database to deal with the information due to the fact it's far continuously expanding.

Types of NoSQL Databases

There are numerous extraordinary sorts of NoSQL databases, and the call of the database device that suits that class is:

1. Graph Databases: Examples – Amazon Neptune, Neo4j
2. Key fee save: Examples – Memcached, Redis, Coherence
3. Tabular: Examples – Hbase, Big Table, Accumulo
4. Document-primarily based totally: Examples – MongoDB, CouchDB, Cloudant

Features of NoSQL Databases

1. The dynamic schema of NoSQL databases permits them to modify to evolving statistics systems with out the requirement for migrations or schema adjustments.
2. Horizontal Scalability is supported. NoSQL databases can manage huge quantities of statistics and excessive site visitors given that they're designed to scale up through including extra nodes to a database cluster.
3. A range of NoSQL databases, which includes MongoDB, keep statistics in semi-based paperwork like JSON or BSON the usage of a document-primarily based totally statistics model.
4. The key-fee statistics model, which Redis and different NoSQL databases employ, shops statistics as a set of key-fee pairs.
5. A column-primarily based totally statistics model, used by a few NoSQL databases like Cassandra, arranges statistics into columns instead of rows.
6. Distributed and excessive availability: NoSQL databases are regularly made to be quite to be had and to attend to statistics replication over severa nodes in a database cluster in addition to node failures.
7. With guide for severa statistics kinds and dynamic statistics systems, NoSQL databases provide builders the liberty to keep and retrieve statistics in a bendy and dynamic manner. It Provides Flexibility.

Benefits of NoSQL

There are severa benefits to running with NoSQL databases like MongoDB and Cassandra. The most important benefits are scalability and excessive availability. The question language is supported. It provides short performance, permits for horizontal scaling.

High scalability: In NoSQL databases, shuddering is used for horizontal scaling. Data may be partitioned and disbursed over a couple of machines at the same time as maintaining its authentic order the usage of a system referred to as sharding. While horizontal scaling is simple to do, vertical scaling is extra complicated. Examples of databases with horizontal scaling are MongoDB, Cassandra, and others. Scalability is one of the motives NoSQL is capable of manipulate big volumes of statistics. NoSQL is able to coping with developing volumes of statistics.

Flexibility: NoSQL databases are designed to address unstructured or semi-based statistics, and they'll modify to slow adjustments withinside the statistics model.

For software program programs that want to manipulate converting statistics requirements, NoSQL databases are an first-rate alternative due to this.

High availability: Because the information mechanically replicates to the maximum latest solid country withinside the case of a failure, NoSQL databases are particularly available.

Scalability: NoSQL databases are very scalable, which permits them to without difficulty take care of huge quantities of information and traffic. As a end result, they paintings properly for programs that need to control huge quantities of information or traffic.

Performance: NoSQL databases are designed to deal with huge volumes of information and traffic, they are able to carry out higher than conventional relational databases.

Cost-effectiveness: NoSQL databases are usually much less steeply-priced than conventional relational databases due to their less difficult layout and shortage of luxurious hardware or software program requirements.

1. Disadvantages of NoSQL Databases

1. There are many special forms of NoSQL databases, every with specific blessings and drawbacks. The loss of requirements in a selected utility could make it hard to pick out the proper database.
2. Lack of ACID compliance: Consistency, integrity, and sturdiness of information aren't assured in NoSQL databases considering the fact that they do now no longer completely adhere to the ACID standard. This can be a downside for programs that require excessive ensures of information consistency.
3. Narrow Focus: Since a massive part of NoSQL databases` capability is dedicated to storage, their attain is alternatively constrained. Relational databases are higher than NoSQL withinside the subject of transaction management.
4. Open-source: The NoSQL database is loose to use. There is not but a honest NoSQL standard. To positioned it some other way, database structures are possibly now no longer equal.
5. Lack of assist for complex queries: Because NoSQL databases are not constructed to deal with them, they are now no longer a appropriate healthy for programs that want complicated information evaluation or reporting.
6. Huge record size: Many database structures, which includes CouchDB and MongoDB, save information withinside the JSON format. This shows that the scale of the files is reasonably vast (in phrases of BigData, community capacity, and speed), and the descriptive key names in reality motive the files to get larger.
7. Lack of maturity: NoSQL databases are nevertheless growing and aren't as advanced as traditional relational databases. They can be much less reliable and steady as a end result than traditional databases.
8. Backup: Certain NoSQL databases, like MongoDB, have a critical weak spot on the subject of backup. There isn't anyt any approach for constant information backup in MongoDB.

9. Management difficulty: The intention of massive information gear is to simplify the method of handling massive quantities of information. However, it is now no longer that easy. The information management in NoSQL databases is drastically extra complicated than in conventional databases. It's widely known that NoSQL specially is notoriously tough to put in and extraordinarily tougher to frequently maintain.

10. There isn't anyt any GUI: Market gear for having access to databases in GUI mode aren't extensively available.

1.5 Multimedia Databases

A series of associated multimedia facts, such as text, graphics (sketches, drawings), images, animations, video, audio, and different facts kinds from many sources, is called a "multimedia database". A framework for arranging extraordinary styles of multimedia facts that can be supplied, saved, and utilized in numerous approaches is known as a multimedia database control system. Dimensional media, dynamic media, and static media are the 3 instructions that make up the multimedia database.

Multimedia Database Management System Content

1. Media facts -Actual facts used to depict an item is known as media facts.
2. Media layout Data-Information regarding the layout of the media facts after it has passed through the acquisition, processing, and encoding stages, consisting of sampling charge, resolution, encoding method, etc.
3. Media key-word facts: Keywords that describe how facts is generated. It additionally is going with the aid of using the call of "content material descriptive facts. "Example: the recording`s date, time, and location.
4. Media function facts -The distribution of colours, styles of texture, and numerous shapes which are found in facts are examples of media function facts, that's content material-established facts.

Many multimedia apps may be categorised in keeping with how they cope with facts as follows:

Repository application: Large volumes of multimedia facts are saved in what are known as repositories, collectively with meta-facts (such media layout date, media key-word, and media function facts) for next retrieval. Satellite photos, radiography scanned images, and engineering drawings are some examples of repositories.

- Applications for presentations: With these, multimedia facts distribution is restricted with the aid of using time. The DBMS should deliver facts at a selected charge at the same time as retaining a stage of provider above a predetermined threshold so as for viewing or paying attention to be optimal. Data is processed right here as it's far being supplied. As an illustration, don't forget modifying evaluation in realtime with video and audio facts.
- Collaborative paintings the use of multimedia records: Working collectively to finish a tough task with the aid of using combining drawings and changing notifications is called collaborative paintings with multimedia records. For instance, an smart healthcare network.

Multimedia databases are used withinside the following areas:

- Sectors and corporations that cope with and preserve certain facts and loads of papers. For instance: An coverage declare record.
- Knowledge dissemination: A multimedia database, which presents a number of resources, is a completely beneficial device for know-how dissemination. Take digital books, for instance.
- Education and training: Computer-aided mastering merchandise may be created the use of multimedia sources, which can be nowadays very famous mastering tools. Take virtual libraries, for instance.
- Shopping, traveling, having fun, and product promotion. An instance is probably an internet metropolis guide.
- Monitoring and manipulate in actual time: Multimedia records display, whilst paired with lively database technology, may be a relatively beneficial device for coping with and tracking hard tasks. control of manufacturing processes, for instance. Issues with multimedia databases
- Modeling - By improving database retrieval strategies rather than data retrieval techniques, files constitute a selected subject and advantage unique attention.
- Design - The conceptual, logical, and bodily layout of multimedia databases have now no longer but been absolutely addressed when you consider that overall performance and tuning worries at every degree are a good deal greater complex due to the fact they

contain of various codecs like JPG, GIF, PNG, and MPEG which can be tough to transform between.

- Storage - When a multimedia database is saved on a everyday disc, representation, compression, mapping to tool hierarchies, archiving, and buffering throughout input-output operations grow to be issues. Untyped bitmaps may be stored and retrieved the usage of a DBMS function called "BLOB" (Binary Big Object).
- In packages regarding audio-video synchronization or video playback, overall performance is ruled with the aid of using bodily limits. While those strategies are nevertheless of their early tiers of development, the usage of parallel processing should assist with a few problems. Multimedia databases additionally require a whole lot of bandwidth and computing power.
- Queries and retrieval: Using a question to retrieve multimedia facts, which include pictures, videos, and audio, offers lots of demanding situations that have to be overcome, inclusive of green question creation, green question execution, and green question optimization.

1.5Big Data Databases

- Big Data is a group of data this is continually developing swiftly in quantity. The quantity and complexity of this facts make it not possible for traditional facts control answers to deal with or store. Big facts is a time period used to explain very massive facts sets. Big facts is characterised with the aid of using more variety, quicker arrival times, and better quantity of facts. Another time period for that is the 3 Vs.Big Data is described with the aid of using 3 factors: quantity, diversity, and velocity. These characteristics together characterize "Big Data." Big facts is only a time period for larger, greater complex facts collections, specially from new sources. Big facts is a time period used to explain very massive facts sets.Example of Big Data are Amazon Redshift, Azure Synapse Analytics, Microsoft SQL Server, Oracle Database, MySQL, IBM DB2, etc.

Sources of Big Data

- Social networking webweb sites: Social networking webweb sites, which include Facebook, Google, and LinkedIn, produce substantial quantities of facts each day because of their big person bases.

- E-trade website: Sites like Amazon, Flipkart, and Alibaba produce a widespread quantity of logs that may be used to tune patron shopping styles.
- Weather station: All climate stations and satellites offer substantial quantities of facts, which might be then saved and processed to forecast the climate. Because they've billions of participants globally, social networking webweb sites like Facebook, Google, and LinkedIn all produce substantial quantities of facts each day.
- E-trade website: Websites like Amazon, Flipkart, and Alibaba produce a widespread wide variety of logs from which patron shopping styles may be identified.
- Weather station: Every climate station and satellite tv for pc affords a ton of facts this is saved and processed to forecast the climate.
- Telecom company: Telecom behemoths like Airtel and Vodafone studies client developments after which announce their plans in step with the ones findings. To do this, they keep the facts in their million subscribers.
- Share Market: Through their ordinary transactions, inventory exchanges all around the international produce a enormous quantity of facts.
- 3V`s of Big Data
- • Velocity: The facts is increasing at a totally speedy rate. Every years, the extent of facts is expected to double.
- Variety: Today's facts are not saved in rows and columns. Data is probably prepared or unstructured. Unstructured facts consists of log documents and CCTV footage. Tables may be used to keep established facts, along with transaction facts from the bank.
- Volume: The petabytes of facts that we deal with constitute an huge quantity of facts.

Types of Big Data

The following are the forms of Big Data:

1.Structured

- Any facts that may be retrieved, processed, and saved in a fixed way is considered established facts. Structured facts may be visible in a database's "Employee" desk.

2.Unstructured

- All facts that may be retrieved, processed, and saved in a fixed layout is known as established facts. Example: The consequences of a "Google Search"

3.Semi-established

- Semi-established facts may be used with each varieties of facts. Semi-established facts seems established, however it is now no longer defined, say, via way of means of a relational database control system's desk definition.An XML record containing a facts set is an instance of semi-established facts.

Advantages of Big Data Processing

- Businesses can leverage outdoor intelligence to tell their decision-making. Thanks to social facts from engines like google and web sites like Facebook and Twitter, companies can also additionally now fine-song their enterprise strategy.
- Improved customer services: Conventional client comments techniques are being changed via way of means of new structures advanced the usage of Big Data era. These new equipment are utilising massive facts and herbal language processing era to examine and compare person input.
- Improved operational efficiency
- Early detection of any damage to the products or services.
- Among the blessings of massive facts consist of higher decision-making, better operational efficiency, and stepped forward client service.

Knowledge Check 2

State True and False for the Following Sentences.

1. One form of little facts is known as massive facts. Big facts is characterised via way of means of extra variety, quicker arrival times, and better quantity of facts.
2. Another time period for massive facts is the Four Vs.
3. A "multimedia database" is a grouping of associated multimedia facts, that may consist of text, images (sketches, drawings), animations, music, video, and different facts from many sources.

Outcome-Based Activity 2

- Describe all forms of Practical Applications of Big Data.

7.7 Summary

- A disbursed database is one wherein no unmarried CPU controls all of the garage devices.
- A parallel database enables to boom overall performance via way of means of parallelizing a whole lot of tasks, along with facts loading, developing indexes, and analyzingquery.
- An object-orientated database (OOD) is a kind of database gadget which can control complicated information gadgets, or gadgets that resemble the ones utilized in object-orientated programming languages. In object-orientated programming (OOP), each aspect is an object.
- Data may be provided and from time to time saved in XML layout the use of an XML database, that is a software program answer for information persistence. It is feasible to query, modify, export, after which go back this information to the caller gadget.
- A subset of document-orientated databases, which in flip are a subset of NoSQL databases, are XML databases.
- NoSQL, additionally noted as "now no longer without a doubt SQL" or "non-SQL," is a technique of designing databases that allows the storing and querying of information out of doors of the standard systems observed in relational databases.
- A "multimedia database" is a group of related multimedia information that could incorporate text, graphics (sketches, drawings), photos, animations, video, audio, and different information kinds from diverse sources.
- Big information is characterised via way of means of a larger variety, better volume, and quicker velocity of information arrival. Another time period for that is the 3 Vs. Big information best refers to larger, extra elaborate information sets, specifically whilst they arrive from clean sources.

7.8 Self-Assessment Questions

1. What is a Parallel and Distributed Database?
2. Explain Object-Based Database.
3. What is XML Database?
4. What is NoSQL Database?
5. Explain Multimedia Database.
6. What is Big Data?
7. Explain the 3 v`s of Big Data.
8. What is Big Data Processing?

7.9 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 8

INTRODUCTION TO DISTRIBUTED DATABASES

Learning Outcomes:

- Students will be capable of learning about the Transparencies in a distributed DBMS.
- The Distributed Database Management System Architecture will be clear to the students.
- Knowledge of Alternative Design Strategies and Global Directory Issues will be attained by the students. Students will be able to understand the Distributed design issues.
- Students will be able to learn about Fragmentation and Data allocation.
- Students will learn about view management and Data Security.
- Students will understand the concept of Semantic Integrity Control.

Structure

8.1 Transparencies in a distributed DBMS

The requirement for a DBMS device to offer the person with a obvious distribution—that is, to shield them from implementation details—is referred to as transparency in DBMSs. Distribution transparency, transaction transparency, overall performance transparency, and DBMS transparency are the 4 classes below which transparency is divided. Transparency in disbursed database control structures pertains to how effortlessly data is transferred from the device to the person.

It facilitates to hide the data that the person have to use. Assume that during a wellknown database control device (DBMS), facts independence is a form of transparency that aids in concealing from the person modifications to the definition and business enterprise of the facts. But all of them have the identical ordinary cause in thoughts. In different words, deal with the disbursed database as you'll a centralised database.

There are 4 specific sorts of transparency in disbursed database control structures, and they may be as follows:

1. Transaction transparency
2. Performance transparency
3. DBMS transparency
4. Distribution transparency

1.Transaction Transparency

Due to this transparency, disbursed databases` integrity and regularity are maintained all through all transactions. It is likewise vital to understand that distribution transaction get right of entry to refers to facts held throughout many places. Another issue to preserve in thoughts is that the DDBMS is in price of making sure that every sub-transaction is atomic, because of this that that it both completes immediately or now no longer at all. Because of the DBMS's use of fragmentation, allocation, and replication, it's miles pretty complicated. One DDBMS function that ensures database transactions will maintain the consistency of the disbursed database is transaction transparency. Data saved at a couple of places is accessed through a disbursed transaction. Transparency in transactions ensures that the transaction will best be finalized while every of the taking part database web web sites has completed their part of the transaction.

2.Performance Transparency

A DDBMS have to characteristic as a centralised database control device a good way to acquire this transparency. Also, due to its disbursed architecture, the device should not revel in any overall performance drops. A disbursed question processor, much like this, is needed with the aid of using a DDBMS to translate a facts request into an orderly collection of sports at the neighborhood database. The fragmentation, replication, and allocation systems of DBMS upload a further layer of complexity to this, which have to be taken into account.

Durability, or all modifications which have been stored completely in a database however have to now no longer be misplaced withinside the case of any type of failure, is a demand for overall performance transparency in a transaction. Accessing and updating facts storages positioned in severa places have to be made extra complicated thru disbursed transactions.

3.DBMS transparency

This transparency is most effective relevant for numerous varieties of Distributed DBMS because it hides the opportunity of a awesome neighborhood DBMS (Databases with numerous webweb sites and the usage of numerous running systems, products, and facts models). One of the maximum tough generalisations is the generalisation of this transparency.

4.Distributed Transparency

a. When a DDBMS helps distribution facts transparency, the consumer isn't required to be conscious that the facts is fragmented and may as a substitute see the database as a unmarried item or logical entity.
b. The 5 varieties of distribution transparency are indexed below.
c. Fragmentation Transparency-Database accesses are primarily based totally on the worldwide schema due to the fact there's no requirement for the consumer to be aware about fragmented facts on this type of transparency. Like SQL views, the consumer can be unaware that they may be the usage of a view of a desk as opposed to the desk itself. This distribution transparency stage is the highest. The characteristic permits customers to question any desk as aleven though it has already been divided up. The reality that the tables being retrieved are, in reality, a fraction or a union of numerous fragments is

concealed. Stated differently, the consumer is blind to the fragmentation of the tables. Because of this, fragment names and facts places aren't required to be special via way of means of the consumer due to the fact database accesses are all primarily based totally on worldwide schema.

d. Location Transparency-Database accesses are primarily based totally on the worldwide schema because of the dearth of requirement for the consumer to be **aware about** fragmented facts on this shape of transparency.The consumer can be blind to the **reality** that they may be using a desk view as opposed to the desk itself, as with the SQL views. The intermediate diploma of distribution transparency is region transparency. It ensures that any tables or fragments may be queried via way of means of the consumer simply as though they have been regionally saved on their site. The consumer wishes to be aware about how the facts has been divided while there's this stage of transparency. They are blind to the facts`s region, nevertheless. A database control system (DBMS) has to have get admission to to an up to date date dictionary and a listing containing the region info of facts so that it will put in force region transparency.

e. Replication transparency- The consumer is blind to fragment copying in replication transparency. Transparency with inside the replication is related to concurrency and failure transparency. When a consumer modifies a selected facts item, the extrade isalso pondered in all desk copies. Yet, the consumer need to be blind to this operation.

f. In replication transparency, the consumer is blind to the fragments replication. It allows customers to question upon a desk as though most effective a unmarried replica of the desk exists. Replication transparency is related to concurrency transparency and failure transparency. When a consumer updates a facts item, the replace is pondered in all copies of the desk. Replication transparency is likewise implied via way of means of region transparency. Local mapping transparency- In order to avoid duplications, the user must define both the fragment names and the positioning of the data items in the local mapping transparency. This is a more difficult and time-consuming query for the user in terms of Distributed DBMS transparency**.** the least transparent distribution level. A user must describe the location of data items as well as the names of the fragments in this type of transparency, taking into account any potential duplication.

g. Naming transparency- Both DBMS and Distributed DBMS are centralized database management systems. In other words, each item in this database must have its own name.

Distributed DBMS must also verify that no database objects with the same name are created by two sites. To address the issue of naming transparency, there are two options: build a central name server to produce unique names for system objects or add an item that starts with the creator site's identity.

8.2 Distributed DBMS Architecture

Applications can access data from databases that are located both locally and remotely thanks to distributed database design. In a homogenous distributed database system, every database has an Oracle database. There is at least one database that is not an Oracle database in a heterogeneous distributed database system. In this two-tier design, the functionality of the servers and clients varies. There are sections for the client and the server. The primary duties of the server are data administration, query processing, optimization, and transaction management. Client functions are the main areas where the user interface is used.

Data management, query processing, optimization, and transaction administration are among the main server tasks. The most used client function is the user interface.

Distributed DBMS Architecture

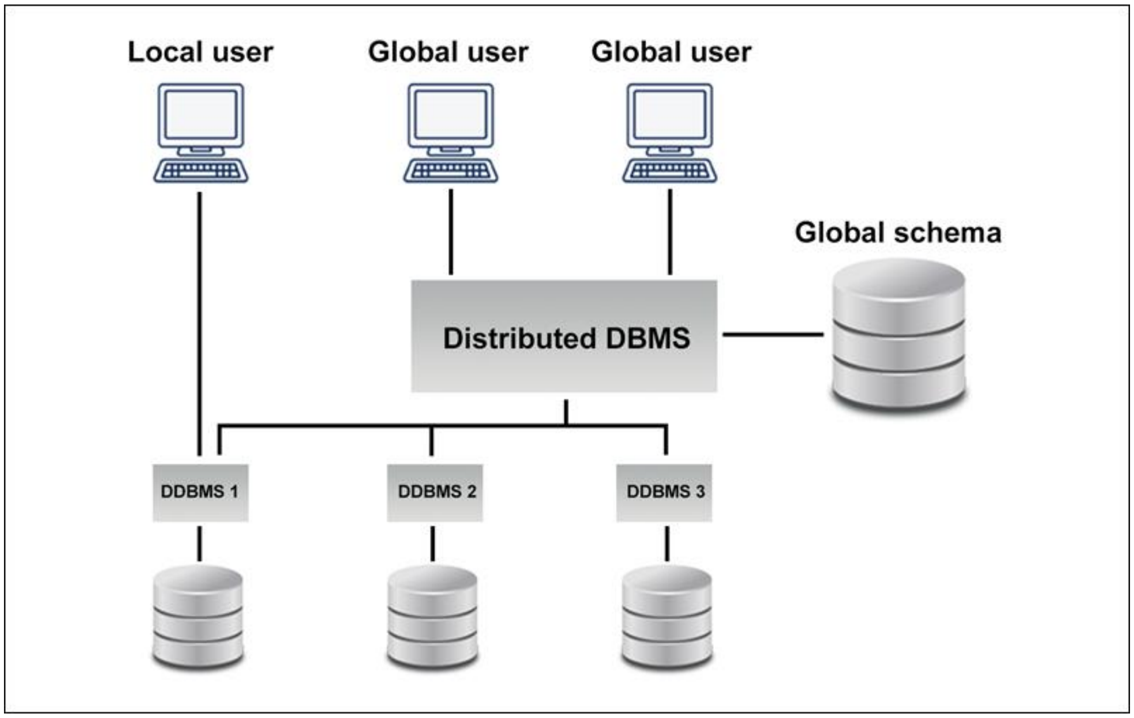


Fig 8.1 Distributed DBMS Architecture

Distributed DBMS architectures are typically designed in the three parameters that are mentioned below as follows:

1. **Distribution -**Distributions used for describing the dispersion in physical form of the various data across distinct sites.
2. **Autonomy** - The spread of database system control and the extent to which each component DBMS can function independently are referred to as autonomy.
3. **Heterogeneity**- It has to do with how databases, system parts, and data models are comparable or different.

Some main architectural models are as follows:

a. Client-Server DDBMS Architecture
b. Peer-to-Peer DDBMS Architecture
c. Multi - DBMS Architecture

a. Client-Server DDBMS Architecture

Servers and clients have different levels of functionality in this two-tier architecture. Data management, query processing, optimization, and transaction administration are among the main server tasks. The most used client function is the user interface. They do, however, offer a few services, like transaction management and consistency checking.

i. It is one of the two unique client-server designs
ii. Multiple Clients, Multiple Servers

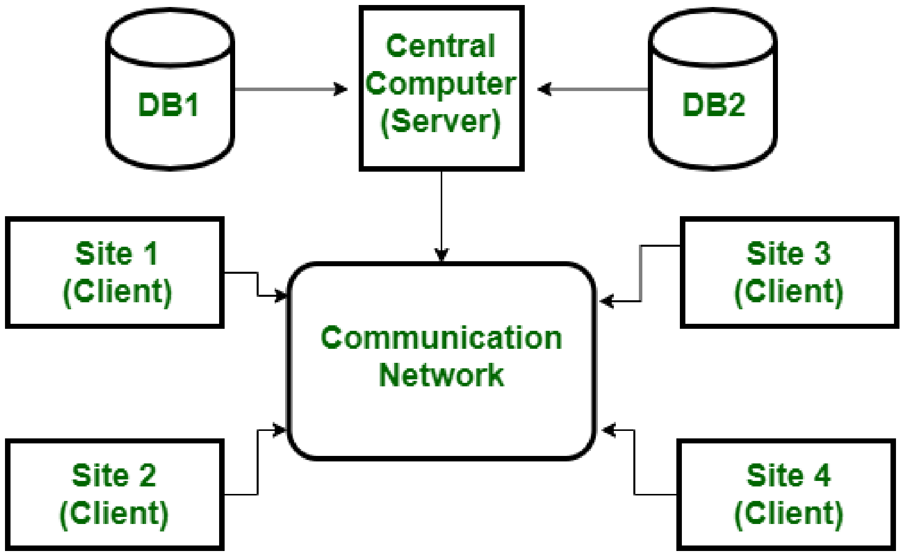


Fig 8.2 Client-Server DDBMS Architecture

b. Peer-to-Peer Architecture for DDBMS

In these types of systems, each peer serves as both a client and a server for providing the database services. Peers pool their resources and collaborate to coordinate their actions. In general, this architecture has four levels of schemas.

i. Global Conceptual Schema: This shows the data's general logical structure.
ii. Local Conceptual Schema: This diagram will display each location's logical data organization. Local Internal Schema: This will display the actual physical configuration of the data at each place.
iii. External Schema: This will show how the user views the data.

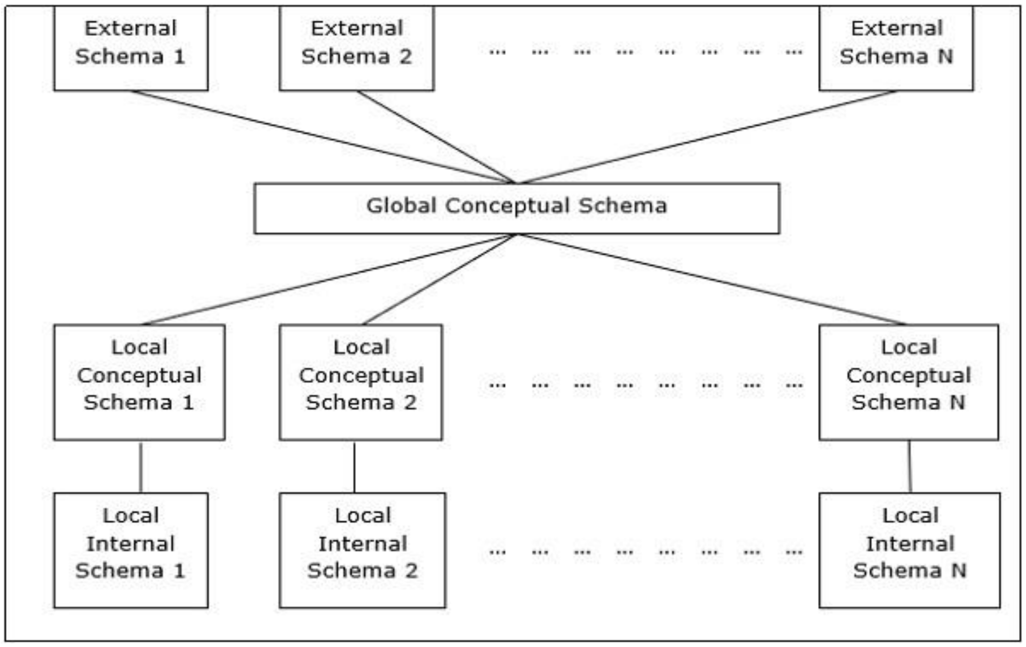


Fig 8.3 Peer-to-Peer Architecture for DDBMS through user views

c. **Multi - DBMS Architectures**

An incorporated database gadget consists of or extra impartial database systems.

Multi-DBMS may be described the use of six stages of schemas.

i. Multi-database View Level– It depicts some of person perspectives which might be subsets of the incorporated disbursed database.
ii. Multi-database Conceptual Level–It is used to explicit a multi-database incorporated shape composed of worldwide logical multi-database shape specifications.
iii. Multi-database Internal Level- It presentations information distribution throughout numerous web-web sites in addition to multi-database to neighborhood information mapping.
iv. Local database View Level–It represents the general public view of the neighborhood information.
v. Local database Conceptual Level–It presentations the agency of neighborhood information at every site.
vi. Local database Internal Level– It depicts the bodily shape of the information at every location.

There are layout opportunities for multi-DBMS, which might be as follows:

a. Model on the conceptual stage of many databases.
b. The model`s conceptual stage within side the absence of a multi-database.

8.3 Global Directory Issues

It includes details regarding the composition and location of the fragments. With meta-data describing the actual data kept in the database, the directory functions as a standalone database.

A global conceptual schema is used by remote DBMSs and multi-DBMSs. Global Directory is an addition to the traditional directory that contains details on the composition and location of the fragments.

- **Global Directory Issues**

a. Applicable to allotted DBMS or multi-DBMS that use a international conceptual schema.
b. Includes statistics approximately the place and composition of the fragments.
c. The listing is a database in its very own right, containing meta-records approximately the real records saved within side the database.
d. A listing is probably international to the complete database or site-specific.
e. The listing can be stored centrally in a single place or allotted over several locations.
f. The listing is continually allotted if the gadget is allotted.
g. Replication may be completed in a unmarried or a couple of copy. Multiple copies might improve reliability.

8. Four Alternative Design Strategies

The distribution layout opportunities for tables in a Distributed DBMS are as follows:

a. Non-replicated and non-fragmented
b. Fully replicated
c. Partially replicated
d. Fragmented

e. Mixed

a. Non-replicated &Non-fragmented

Several tables are located in diverse positions on this layout variant. Data is positioned close to the place wherein it's far maximum regularly used. It is satisfactory ideal for database structures while the percentage of queries required to enroll in records from disparate tables is low. This design choice contributes to minimize transmission costs during data processing if a suitable distribution strategy is utilized.

b. Fully Replicated

Each site in this architecture choice holds a single copy of all database tables. Queries are highly fast and incur negligible communication costs because each site has its own copy of the whole database.

In contrast, significant data redundancy needs extravagant expenses during update operations. As a result, this is performed for systems that will must process a large count of requests while performing a small number of database updates.

c. Partially Replicated

You can find copies of tables or parts of tables anywhere. The tables are set up according to how frequently they are accessed, descending. This accounts for the fact that different sites consult the tables at very different frequencies. The frequency of access queries and the website that generates the queries decide how many copies of the tables—or sections of the tables—are made.

d. Fragmented

This structure divides a desk into or extra quantities or partitions, and every fragment may be retained in separate locations. This bills for the truth that it's far unusual for all statistics recorded in a desk to be required at a unmarried location. In addition, fragmentation complements parallelism and catastrophe recovery. Each fragment in this example has simplest one replica within side the gadget, ensuing in no redundant statistics.

The 3 fragmentation strategies that are given underneath as follows−

1. Vertical fragmentation
2. Horizontal fragmentation
3. Hybrid fragmentation

e. Mixed Distribution

Due to fragmentation and partial replications, Mixed Distribution occurs. The tables are first fragmented in any orientation (horizontal or vertical), and the portions are then partly reproduced throughout a couple of web-web sites primarily based totally on how regularly the fragments are accessed.

8. **Five Distributed Design Issues**

The vital layout worries are fragmentation (the separation of the database into fragments) and distribution (the fine distribution of portions). A disbursed facts gadget is described as "a set of networked computer systems related collectively through a community with the intention of sharing facts amongst them. "A disbursed facts gadget is made from several impartial computer systems that talk with each other and percentage statistics through a pc community.

1. Heterogeneity: This phrase refers to extraordinary pc hardware, running structures, networks, and implementations utilized by developers. A key detail of the client-server ecology of heterogeneous disbursed structures is middleware. Middleware is a set of offerings that enables communiqué among programs and cease customers in a heterogeneous disbursed gadget.
2. Openness: The openness of a disbursed gadget is decided via way of means of the convenience with which new resource-sharing offerings may be made to be had to customers. The truth that open structures` vital interfaces are posted units them apart. It is primarily based totally on a well known communique protocol and a publicly on hand

interface for getting access to shared assets. It may be built the usage of a extensive variety of hardware and software.

3. Scalability: The gadget's scalability need to continue to be green even if the range of customers and assets related grows significantly. It need to make no distinction whether or not a programme has 10 or one hundred nodes; overall performance need to be consistent. Scaling a disbursed gadget necessitates attention of numerous factors, such as size, geography, and administration.
4. Security: Information gadget safety has 3 components. Confidentiality, integrity, and accessibility are all crucial considerations. Encryption safeguards shared assets and disguise touchy facts even as it's far transmitted.
5. Failure Handling: It is important to set up corrective techniques to address the situation whilst hardware and software program malfunctions result in misguided outcomes or halt earlier than finishing the supposed computation. Because partial screw ups in dispensed structures arise whilst a few additives fail at the same time as others maintain to operate, handling those types of screw ups may be challenging.
6. Concurrency: Concurrency takes place whilst several customers try to get admission to the equal shared useful resource on the equal time. Many customers make requests to read, write, and replace the equal resources. Each useful resource in a concurrent context need to be safe. Any item representing a shared useful resource in a dispensed gadget need to make sure that it plays effectively in a concurrent environment.
7. Transparency: By being obvious, a dispensed gadget is perceived as a unmarried entity through customers or utility programmers, instead of a set of unbiased entities that collaborate. The changeover from a neighborhood to a far off computing device ought to be seamless, and the person ought to be oblivious of wherein the offerings are located.

• Knowledge Check 1

Fill within side the Blanks

1. _____________ in DBMS refers back to the requirement that a DBMS gadget affords the person with a obvious distribution
2. __________________ is an addition to the conventional listing that consists of information about the pieces` places and composition.
3. ____________tiers of schemas may be used to outline multi-DBMS.

- Outcome-Based Activity 1

Describe the various kinds of structure of the Distributed DBMS.

8.6 Fragmentation

The procedure of breaking a desk up into smaller tables is known as fragmentation. We talk to the desk's subsets as fragments. The procedure of breaking a desk up into smaller tables is known as fragmentation.

The desk's subsets are known as fragments. Three kinds of fragmentation are distinguished: hybrid, which mixes vertical and horizontal elements, and horizontal. Primary and derived horizontal fragmentation are in addition classes into which horizontal fragmentation strategies may be separated. It is important to fragment the desk which will reassemble the unique from the pieces. This is wanted in case it will become important to reconstruct the unique desk the use of the pieces. That's what we call "reconstruction."

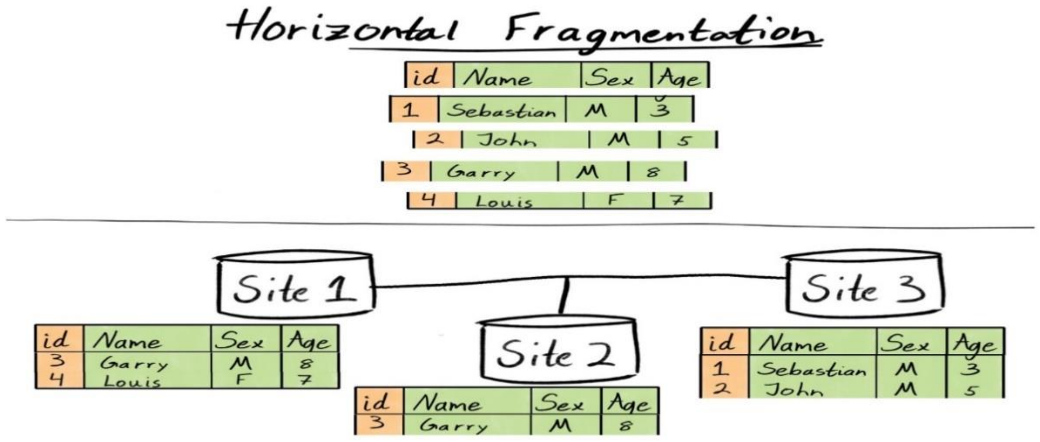


Fig 8.4 Horizontal Fragmentation

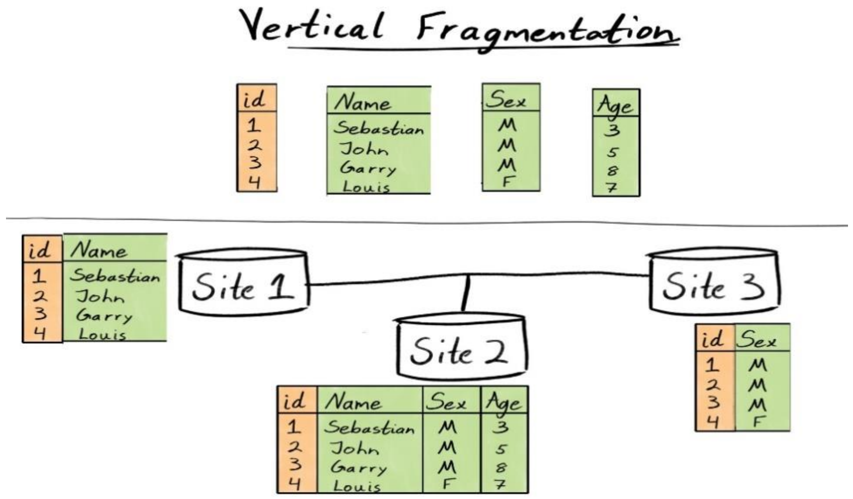


Fig 8.5 Horizontal Fragmentation

Advantages of Fragmentation

1. Local query optimization procedures are sufficient for some queries because the data is available locally.
2. Fragmentation aids in the confidentiality and privacy of the database system.

Disadvantages of Fragmentation

1. Access times may be extremely slow when data from numerous fragments is requested.
2. Rebuilding recursive fragmentations will need the deployment of costly techniques.
3. If a site fails, a lack of backup copies of data on other sites may leave the database inoperable**.**

- **Types of Fragmentation**

1.Vertical fragmentation

A table's fields or columns can be divided into pieces via a process called vertical fragmentation. Every component should have the main table field in order to preserve reconstructive (s). Data privacy can be protected by the use of vertical data fragmentation.

2.Horizontal Fragmentation

Table tuples are organized in a horizontal style in line with the values of 1 or extra fields. Additionally, horizontal fragmentation should uphold the reconstructive norm. All of the columns from the unique base desk should be found in every horizontal segment.

3.Hybrid fragmentation

A hybrid fragmentation method combines the horizontal and vertical fragmentation approaches. Given that it creates fragments with the least quantity of pointless data, this fragmentation method is the maximum adaptable. Unfortunately, there are times whilst duplicating the unique desk is a luxurious process.

Two strategies exist for hybrid fragmentation.

1. Start with the aid of using assembling a set of horizontal pieces, after which use one or extra of the horizontal fragments to assemble vertical pieces.
2. Start with the aid of using assembling a set of vertical pieces, after which use one or extra of the vertical fragments to assemble horizontal pieces.

8.7 Data Allocation

Data allocation is the strategic distribution of your records pieces, additionally called records fragments, to enhance database performance and stop person get right of entry to to records. In your DDBMS structures, it goals to lessen basic transaction processing prices at the same time as concurrently turning in fast, reliable records. One of the primary layout demanding situations of a disbursed database gadget is records allocation, despite the fact that those structures have numerous benefits over centralized database structures. A critical degree within side the introduction of disbursed database structures is records allocation. Two not unusual place techniques for powerful records allocation are records replication and records fragmentation.

The techniques used for disbursed database layout are categorized as follows:

1. Fragmentation: A method for breaking apart a database into logical chunks is known as as fragmentation. Fragments may be assigned to exclusive garage locations. Vertical fragmentation, horizontal fragmentation, and hybrid fragmentation are the 3 forms of fragmentation procedures. The fragmentation method presents numerous benefits, which

include improved performance, nearby question optimization, security, and privacy. In a horizontal fragmentation, every tuple of a relation r is assigned to at least one or extra pieces. Vertical fragmentation divides a relation r`s schema into numerous smaller schemas that percentage a not unusual place candidate key and a selected property.

2. Replication: Data replication is a mechanism that lets in unique records or copies of a database to be stored in a couple of location. It is a famous disbursed database fault tolerance approach. Furthermore, a number of the number one advantages of the records replication method encompass reliability, quicker response, decreased community traffic, and easier transactions. Yet, there are numerous downsides to Distributed Database Replication. To make sure particular and right responses to person inquiries, records should be often up to date and synchronized. Failure to achieve this can bring about records inconsistencies, which may stymie company dreams and decisions made with the aid of using different departments.
3. Allocation: Parts or clones of fragments are allotted for garage at numerous locations the use of the allocation approach.

   All data associated with records fragmentation, allocation, and replication is stored in a international listing. This listing is obtainable to the DDBS apps whilst needed.

8.8 View management

A view in a relational database management system is defined as a "virtual table" derived from a specific query on one or more underlying tables. A view can be defined using relational operations such as join, restrict, and project, as well as statistical summaries of data. Complete logical data independence is provided by View. A query on base relations yields the definition of a view, which is a virtual relation. Usually, views are not realized. The pane is dynamic and shows all significant updates to the database.

The view definition in DDBMS is similar to that of centralised DBMS.A view in a DDBMS, on the other hand, can be derived from fractured relations maintained at separate locations.

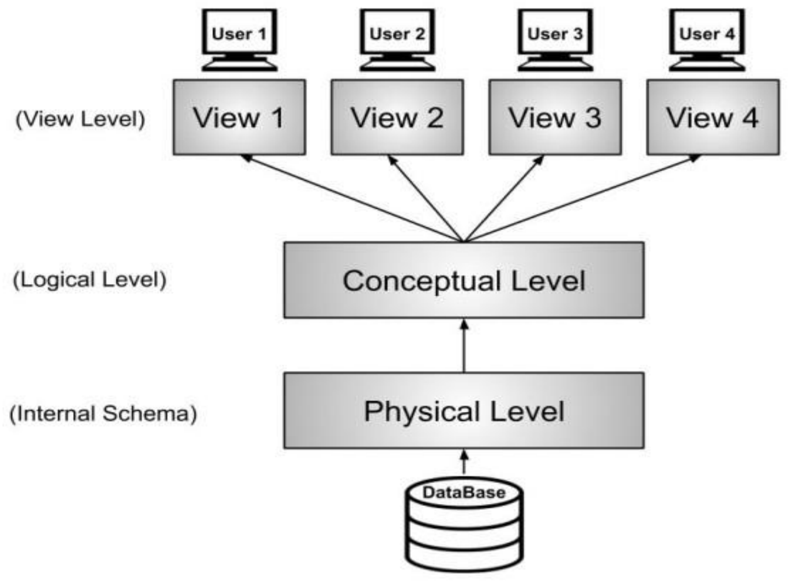


Fig 8.6 Levels of Abstraction

Advantages of View Management

1. It Provides Data Integrity.
2. It helps in proving Logical Data Independence.
3. Views are also value in context of the security.
4. It helps in Shieldingfrom change.
5. It provides Easier querying.

8.9 Data Security

- A disbursed device wishes greater safety features than a centralized one because it has greater customers, varied facts, diverse sites, and disbursed manage.
- In this chapter, we can observe the diverse aspects of disbursed database protection.
- In disbursed verbal exchange systems, intruders fall into types:
- 1. Sensitive data is received through passive eavesdroppers thru message watching.
- 2. In addition to retaining a watch on messages, lively attackers actively make a contribution new facts or regulate present facts.
- Security measures consist of communications protection, facts protection, and facts auditing.

Communications Safety

- Due to the various region of facts, customers, and transactions in a disbursed database, a superb deal of facts verbal exchange occurs. As a result, it necessitates secure verbal exchange among customers and databases, in addition to among database environments.
- The following components of verbal exchange protection are included:
- Data must now no longer be corrupted all through transfer, and the verbal exchange connection must be steady in opposition to each passive and competitive eavesdroppers.
- Well-described protection techniques and protocols must be hired to fulfill the aforementioned standards.
- End-to-stop steady communications are ensured through not unusual place and regular technologies.
    a. Secure Socket Layer Protocol (SSL) or Transport Layer Security Protocol (TLSP).
    b. Virtual Private Networks (VPN).

Data Security

- a. Implementing facts safety features is important in disbursed systems, other than verbal exchange.
- The following are the facts safety features: a. Access manage tactics like authentication and authorization assure that most effective legal customers can get right of entry to the database. Authentication is hooked up thru virtual certificates. Additionally, most effective a username and password mixture may be used to log in.
- In disbursed systems, facts encryption may be carried out in one of a kind techniques. As a part of the disbursed database architecture, the person apps encrypt the facts earlier than storing it within side the database. The applications retrieve and decrypt encrypted facts from the database earlier than the usage of it. External to the disbursed database device: The encryption abilities are constructed in the disbursed database device itself. Without understanding it, person apps keep and retrieve facts from the database this is encrypted.
- Validated enter - This protection function calls for that every enter be validated through the person software earlier than it could be applied to adjust the database.
- A type of attacks, including buffer overruns, command injection, cross-web website online scripting, and facts corruption, may be initiated through the usage of invalid enter.

•Data Auditing

To determine which security features to install place, a database safety device wishes a good way to hit upon and display safety transgressions. It may be tough to apprehend safety breaches once they occur. One approach for locating safety flaws is to check audit logs.

8.10 Semantic Integrity Control

Semantic integrity guarantees that records entered right into a row conform to a valid row cost. The cost need to be in the column`s area or allowable variety of values. In the gadgets table, for example, the amount column best permits numbers. Semantic integrity manage defines and enforces the integrity constraints of the database device**.**

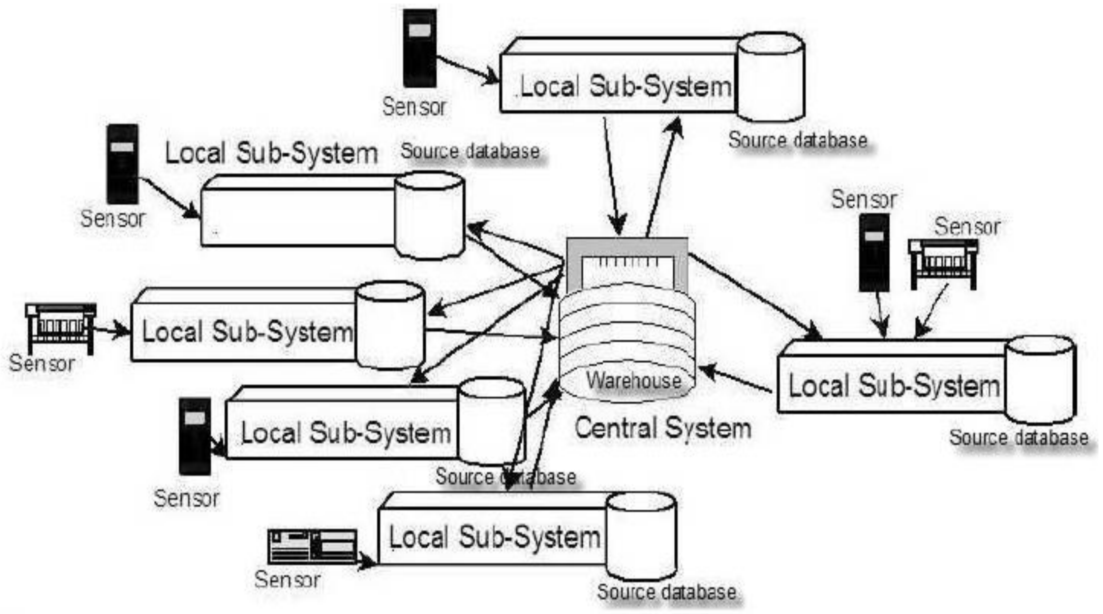


Fig 8.7 Semantic Integrity Control

The following are the integrity constraints:

a. Data type integrity constraint
b. Entity integrity constraint
c. Referential integrity constraint

1. Data Type Integrity Constraint

A data type constraint limits the kinds of operations and values that can be applied to a field of a certain data type. Think about the three fields in the "HOSTEL" table: capacity, name, and number of the hostel. There is a maximum capacity of 150 people, and the hostel number must start with the capital letter "H" and cannot be NULL.

2. Entity Integrity Control

The regulations are upheld via entity integrity control, enabling each tuple to be distinguished from the others. A primary key is defined for this. A primary key comprises the bare minimum of fields necessary to uniquely identify a tuple. No two tuples in a table can have the identical primary key value, and no primary key field can have a NULL value, according to the entity integrity criterion.

3. Referential Integrity Constraint

Foreign key constraints are established by the referential integrity constraint. A foreign key is a field in one data table that is the primary key in another. According to the referential integrity constraint, the value of the foreign key field must either be one of the values of the primary key of the referenced table or be absolutely NULL.

- **Knowledge Check 2**

  State True and False

  1. Replication technique for breaking up a database into logical chunks is called as fragmentation.
  2. A view management stores all data fragmentation, allocation, and replication information. The DDBS apps can access this directory as needed.
  3. The referential integrity constraint establishes the restrictions for foreign keys.

- **Outcome-Based Activity 2**

  Given the following relation EMP and the predicates p1: SAL > 23000, p2: SAL <23000

ID	Name	Salary
1289	Guru	12000
8907	Siva	67050
7643	Shalini	51980
0988	Kavin	23000
6543	Surya	28760
0986	Kavitha	23000
2345	Anees	29999

Perform a horizontal table fragmentation based on the predicates provided.

8.11 Summary

- Transparency in DBMS refers to a DBMS gadget that gives a obvious distribution to the consumer. In different words, it shields the consumer from implementation details. Transparency is assessed into 4 types: distribution transparency, transaction transparency, overall performance transparency, and DBMS transparency.
- The international listing Contains data approximately the fragments` region in addition to their composition. The listing is a database in and of itself, such as meta-information approximately the actual information stored within side the database.
- Fragmentation is the procedure of breaking a desk up into smaller tables. The subsets of the desk are known as fragments.
- Fragmentation is available in 3 flavors: hybrid, which mixes vertical and horizontal elements, and horizontal.
- The primary goal of information allocation in a dispensed database is to distribute information over numerous web-web sites in a manner that minimizes the general value of information transmission at the same time as carrying out some of queries.
- Data safety protocols are positioned into place. To ensure that simplest human beings with permission can get right of entry to the database, get right of entry to manipulate procedures like authentication and authorization are employed. Authentication is showed

with virtual certificates. Also, simplest a username and password aggregate may be used to log in.

- Semantic integrity assures that information located right into a row displays a suitable price for that row. The price need to fall within side the domain, or permissible set of values, for that column. For example, the amount column within side the objects desk simplest accepts numbers.

8.12 Self-Assessment Questions

1. What are Transparencies in dispensed DBMS?
2. Explain DBMS architecture.
3. What are Global listing issues?
4. What are Alternative layout strategies?
5. Explain Distributed layout issues.
6. What are Fragmentation and Data allocation?
7. What is view management?
8. What is Semantic Integrity Control?

8.13 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.

UNIT 9

QUERY PROCESSING IN DISTRIBUTED DATABASE

Learning Outcomes

- Students will be capable of learning about the Objective so query processing.
- Students will be able to understand the Characterization of query Processors.
- Students will be able to understand the concepts of Layers of query processing.
- Students will be able to understand the Query decomposition.
- Students will learn about the Localization of distributed data.

Structure

9.1 Objectives of query processing

- **Query Processing**

The translation of high-level requests in to low-level phrases is known as query processing. It is a step-by step procedure that may be used to achieve the result at the physical level of the file system, query optimization, and actual query execution. It needs a fundamental understanding of relational algebra and file structure. Query optimization is the concept of determining an efficient execution strategy for processing a query. The query processor is the subcomponent of the data server that processes SQL requests. This requests only access a single database or file system or reference multiple types of databases or file systems.

Query processing is the action of take out data from a database.

In this processing, multiple procedures are used to retrieve data from the database.

The steps are as follows:

1. Parsing and translation
2. Optimization
3. Evaluation

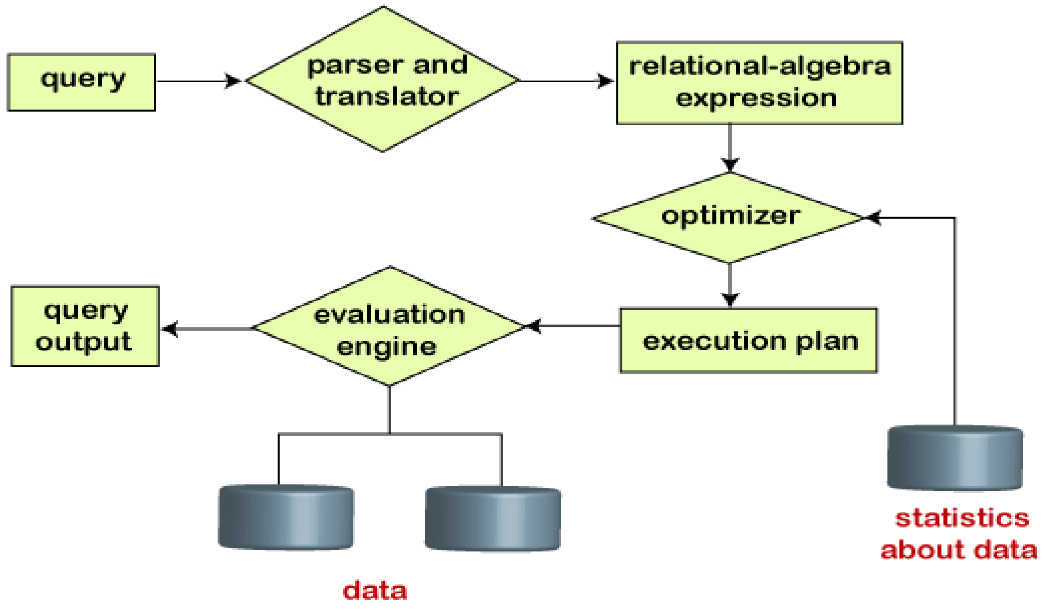


Fig. 9.1 Steps in Query Processing

- **Aim of Query Processing**

The main aim of distributed query processing are to translate good approach query on a distributed database, which users perceive as a single database, into an efficient output generating strategy define din a low-level language in local databases. The goal of query optimization is to find an output generation plan that reduces the time required to run a query. To reach this optimization goal, we must carry out two critical tasks. The first is to determine the most efficient method of accessing the database, and these condis to reduce the time It takes to run the query plan.

a. The purpose of distributed query processing is to translate a high-level query from a distributed database in to a low-level language that can be executed on local databases. The rare several stages to the query transformation.
b. The optimizer attempts to provide the optimal execution plan possible for a SQL statement. The optimizer chooses the plan with the lowest cost amongall candidate plans considered. The optimizer computes expenses based on the information provided. The cost computation takes in to account query output generation factors such as I/O, CPU, and connectivity for a certain query in a given way.
c. As the database has so many internal strategies and devices at its disposal, the optimizer is frequently in a better position to determine the best way to execute a statement than the user. As a result, all SQL statements make use of the optimizer.
d. There is a query, for example, that requests information about students in positions of leadership, such as class reps. If the optimizer statistics show that half of the students are in positions of leadership, the optimizer may decide that a complete tables ear chis the most efficient. However, if data indicate that just a small number of students are in positions of leadership, reading an index followed by database access by row id may be more efficient than a full table scan.

9.2 Characterization of query Processors

The first four features apply to both centralized and distributed query manipulators, where as the next four are exclusive to the distributed query processors in closely integrated distributed DBMSs.

a. Languages
b. Optimization Types
c. Optimization
d. Timing
e. Statistics
f. Sites of Decision
g. Use of Network Topology
h. Replicated Fragment Exploitation
i. Use of Semi joins

1. **Types of Optimization**

   a. **Exhaustive search-** The goal of query optimization is to find the "best" ways in the solution space of all possible output techniques. Search the solution space for the cost of each approach hand select the strategy with the lowest cost. All thought his method is successful for finding he optimum strategy, the optimization it self may entail large processing costs. The problem is that the solution space can been enormous, which means that even with a limited number of relations, there may be numerous comparable techniques.

   b. **Heuristics Search-**Heuristics are a popular meth of decreasing the cost of an exhaustive search by restricting the solution space to a few method that gather common sub-expressions. Do selection and projection first, then replace a join with a sequence of semi-joins, reorder processes to reduce intermediate relation size, and optimize individual operations to minimize data transfer.

   c. **Randomized strategies-** Randomized strategies Pick are all good solution; it may not necessarily be the best of ,but escape the large cost of optimization in particulars of memory and time usage.

2. **Optimization Timing**

Optimization can be complete statically before running the output query or dynamically while it runs.

a. **Static**

Static query optimization occurs during query compilation. As a result, the expense of optimization can be spread a cross numerous query runs. This time frame is suitable for usage with the exhaustive search strategy. As the diameter of a strategy's intermediate relations are unknown until run time, they must be calculated using database statistics.

b. **Dynamic**

Dynamic run time optimization database strategies are not required to calculate the size of intermediate out comes. The fundamental advantage of dynamic query optimization over static query optimization is that the query process or may see the actual sizes of intermediary relations, reducing the likely hood of a wrong choice. The fundamental draw back is that query optimization, an costly job, must be done for every query run. As a result, this method is suitable for adhoc queries.

c. **Hybrid**

It gives the benefits of static query optimisation.The technique is primarily not in motion ,but dynamic query optimization may occur during query run-time if a significant difference between expected and actual sizes of intermediate relations is observed. If the error in estimate sizes exceeds a certain threshold, re-optimize at runtime.

3. **Statistics**

a. The efficiency of query optimization is dependent on database statistics.
b. Statistics are required for dynamic query optimisation to decide what operators should be run working fast.
c. Static query optimisation is substantially more challenging due to the need to calculate the size of intermediary relations using statistical data.
d. Statistics for query optimisation often adding fragment cardinality and size, as well as the size and number of different values for each feature.
e. More extensive statistics, such as histogram so attribute values, are occasionally used to lessen the risk of error.
f. Statistics are kept up to date on a regular basis, which ensures the inaccuracy.
g. When using static optimisation, critical changes in the strategy used to optimize a query may result in query optimisation.

4. **Decision Sites**
   a. **Centralised decision approach**
      The approach that determines the "optimal" time table is generated by a single site. Simpler knowledge of the complete distributed database is also needed to be required.
   b. **Distributed decision approach**
      The distributed decision approach requires just local knowledge to decide the timetable (elaboration of the optimum strategy).
   c. **Hybrid decision approach**
      Onsite makes important decisions that influence the global timetable. Others make local decisions that maximize the local sub-queries.

5. **Network Topology**

Network topology is the physical and logical organization of nodes and links in a computing network. The distributed query optimisation problem is divided into two parts: selecting world-wide execution strategy relay on inter-site communication and selecting each local execution strategy relay on a centralised kind of query processing algorithm.

   a. **Wide area networks (WAN)-** The cost of point-to-point communication will be the most important. It disregards all other cost variables. Local scheduling based on centralised query optimisation.
   b. **Local area networks (LAN)-** This network topology improves parallel execution while increasing communication costs. The broadcasting power of some local are a networks can be successfully leveraged to optimise join operator processing. Algorithms for star networks are unique.

6. **Languages**

Relational DBMS employ relational calculus. Object DBMS employ Object calculus, which is an extension of RDBMS. XML is a data model that uses X query and X Path for storing and transmitting data over the internet. X Path is an xml path medium that is used to utilize queries to pick nodes from an xml document. XQuery extracts and manipulates data from xml documents, relational databases, and Microsoft Office record that support an xml data source.

9.3 Layers of Query Processing

It has a four layers which are given below as follows:-

1. Query Decomposition
2. Data Localisation
3. Global Query Optimisation
4. Distribution Query Execution

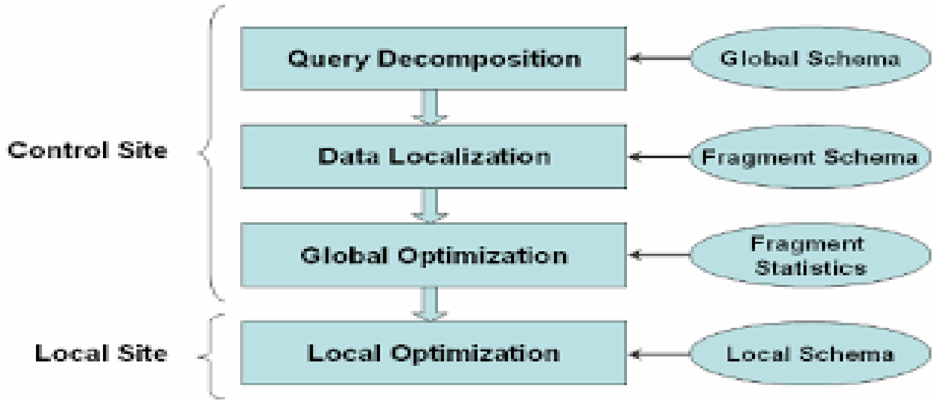


Fig 9.2 Hybrid decision approach

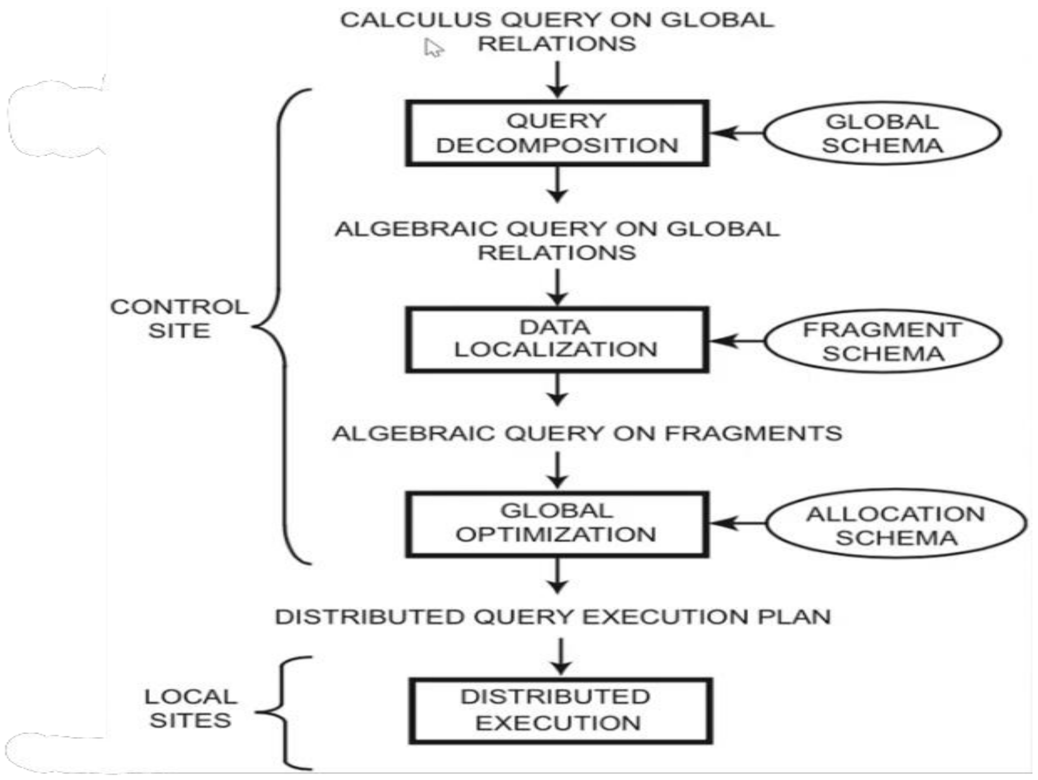


Fig 9.3 Calculus query

1. **Query Decomposition**

The first layer is used for changing the calculus query in an algebraic global relationship query. The data required for this transition is contained in the world-wide conceptual schema reflecting world-wide relations. Query fermentation can be viewed as four successive steps

   a. Normalization
   b. Evaluation
   c. Simplification
   d. Restructure

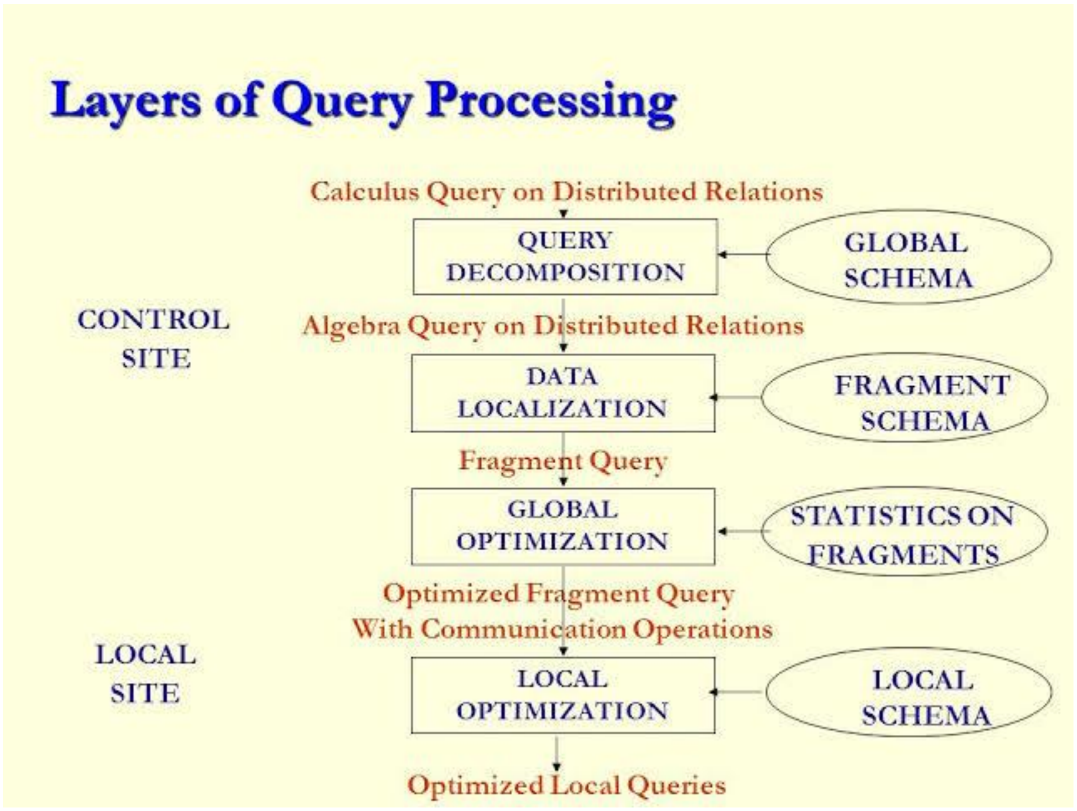
i. The calculus question is first rephrase in a normalized form that can be modified further. Normalising a query frequently requires modifying query quantifiers and query qualification via logical operator priority.
ii. Next, the normalized question is semantically analysed in order to find and reject wrong requests as fast as possible. Techniques for identifying erroneous queries are available in just a subset of relational calculus. They frequently use graphs to capture the semantics of the query.
iii. Next, the right question (it is till expressed in relational calculus) is simplified. Removing super fluous predicates is one method for simplifying a query. It should be noted that when a query is the output of system transformations done on the consumer query, it is likely that redundant queries may occur. These changes are used to govern semantic data (views, protection, and semantic integrity control).
iv. Lastly, the calculus query is reorganized as an algebraic query. The standard type for obtaining "better" algebraic specification is to began with an algebraic inquiry and adjust it to reach" objective."

2. **Data Localisation**

As **input, the next** layer receives an algebraic query on global r**elations. The major goal of this** layer is to localize the query's data utilize.

Data distribution data from the fragment schema. This layer helps to remember which fragments are involved in the query and trans latest to a query on fragments. By apply in the fragmentation criteria and then producing a programme of relational algebra operators that act on parts, a global relation can be restored.

Repetition: Relationships are fragmented, with each kept in a different location. Fragmentation is specified by fragmentation rules or schemes. This layer also detects which fragments are in the query and changes the distributed query into a fragment query. By substituting each distributed relation with its reconstructive programme, the distributed query is mapped into a fragment query (materization program). The fragment query is simplified and rebuilt to another "excellent" query (applying the same rules used in the decomposition layer.)



.

Fig. 9.4 **Layers Of Query Processing**

A fragment query is created in two ways

a. It is first turned into a fragment query by replacing each relation with its recreation programme (also known as the materialization program).

b. The fragment question is then simplified and recreate to generate another "excellent" query.

3. Global Query Optimisation

The third most layer is fed an algebraic query on parts. The aim of query optimisation is to identify a query execution strategy that is close to optimal. Earlier stages have earlier optimized the query, such as by deleting duplicated expressions. This optimisation, however, is unaffected by fragment properties like as fragment all location and cardinalities. The process of obtaining the "optimal" ordering of the query's operators, which includes communication operators that minimise a cost function, is known as query optimisation. The query optimisation layer generates an algebraic query that is optimized and includes communication operators on fragments. It's frequently represented and saved as a distributed query execution plan (for future executions).

Query optimisation is the process of determining the "optimal" ordering of the query's operators, including communication operators that minimise a cost function. The cost function, which is frequently expressed in terms of time units, refers to computer resources like disc space, disc I/Os, buffer space, CPU cost, communication cost, and soon. It is, in general, a weighted combination of I/O, CPU, and communication expenses.Yet, as previously said, an early distributed DBMS simplification was to consider communication cost as the most important criterion. This used to be true for wide-area networks, where restricted bandwidth rendered communication significantly more expensive than local processing. This is no longer the case because communication costs can be lower than I/O costs. To choose an operator or dering, it is important to forecast the execution costs of alternative candidate orderings. Calculating execution costs prior to query execution (i.e., static optimisation) is based on fragment statistics and formulas for predicting the cardinalities of relational operator results. Hence,optimisation decisions are influenced by fragmental location and accessible information on fragments that are recorded in the allocation schema.

Query optimisation consists of

- Identifying the optimum order of operations in the fragment query,
- Identifying the communication procedures that minimise a cost function.
- The cost function refers to computational resource such as discspace, disc I/Os, buffer space,CPU cost,communication cost,and soon.

- Instead of join operators, the semi-join operator isused to optimize queries.

4. Distribution Query Execution

Then layer is processed by every sites that have fragments in the query. Each sub-query executing at a single site, known as a local query, is then optimized and executed using the site's local schema. At this point, the algorithms for performing the relational operations can be selected.

Local optimisational leverages centralised system methods. The purpose of distributed query processing is to identify a suitable execution strategy that minimizes a system cost function that in corporates I/O, CPU,and communication costs given a calculus query on a distributed database. An execution strategy is given in terms of relational algebra operators and communication primitives (send/ receive) added to local databases (i.e., the relation fragments).
As a result, the complexity of relational operators affecting query execution performance is critical in the design of a query processor.

- **I Q Check1**

  Fill the Blank Areas.

  1. __________is the process of determining the "optimal" ordering of the query's operators, which includes communication operators that minimise a cost function.
  2. _____________layer is executed by every sites that have fragmentation in the query
  3. ______________primary function is to localize the query's data utilize data distribution data from the fragment schema.

- **Outcome-BasedActivity1**

  List out all the steps of Query Decomposition.

9.4 Query decomposition

- The query decomposition part of query processing tries to turn a high-level question into a relational algebra questions and to validate that query for syntactic and semantic correctness. The first stage of query processing is query fragmentation. The basic goals of query decomposition are to convert a high-level queries into a relational algebra querries and to validate the query for syntactic and semantic correctness. The typical steps of query fragmentation include analysis, normalisation, semantic analysis, simplification, and query rearrangement.

- The first phase of working on query is query decomposition which converts a relational calculus query into a relational algebra query. Without knowing the distribution of data, both input and output queries pertain to global relations. As a result, question decomposition is identical in centralised and distributed systems.
- The input query is assumed to be syntactically correct in this section. When this stage is completed properly, the output query is semantically correct and good in the sense that it avoids unnecessary labour.
- The following are the steps in question decomposition:
  (1) standardisation,
  (2) analysis,
  (3) redundancy elimination,and
  (4) rewriting.
- Steps1-rely on the notion that for a given query, multiple transformations need to be equivalent, and some may perform better than others. We give the first three phases of tuple relational calculus (e.g.,SQL). The query is only converted to relational algebra in the final phase.

Query fragmentation can be viewed as four steps.

a. Normalisation
b. Analysis
c. Simplification
d. Restructure

Example of Query Processing

Query:

Select salary

From instructor

Where salary<75000;

It can alternatively be expressed as one of the following expressions:

- σsalary<75000(Πsalary(instructor))
- Πsalary(σsalary<75000(instructor))

9.5 Localisation of distributed data

- Data localisaion is the practice of retaining data within the location from when ceit originated. For example, if a company obtains data in the United Kingdom, it will keep it the rather than send it to another country for processing. As input, the second layer receives an algebraic query on global relations.
- The primary and main purpose of the second layer is to localize the query's data using fragment schema data distribution information.
- This layer determines which fragments are involved in the query and converts it to a query on fragments.
- A world-wide relation can be re created by applying there construction criteria and generating a relational algebra programme whose operands are the pieces; this process is known as the localization Program.
- A global relation can be restored by used the fragmentation criterion and then constructing a programme of relational algebra operators that act on pieces.
- The localization layer converts analgebraic query on global relations into an algebraic query on Physical Fragments.
- Localisation takes the use of data stored in the fragment schema. Fragmentation is explained by fragmentation rules, that expressed as relational queries.

Purpose

- It applies data distribution information to algebra operations and assists in determining which fragments are commonly involved.
- It replaces global queries with queries on fragments.
- It improves the global query

- **Knowledge Check 2**

  State true and false for the Following Sentences.

  1. The first step of query processing is query fragementation which converts a relational calculus query into a relational algebra query.
  2. Query decomposition can be viewed a six successive steps.
  3. Data localization is the practice of retaining data within the location from when originated.

- **Outcome-Based Activity2**

Illustrate all the steps of Query Processing.

9.6 Summary

- Query Processing is the process of converting high-level in quiries into low-level expressions. It is a step-by-step procedure that can be utilised at the physical level of the file system, query optimisation, and actual query execution to obtain the result. It necessitates a basic understanding of relational algebra and file structure.
- There are four Layers of Query Processing.
  a. Query Decomposition.
  b. Data Localization.
  c. Global Query Optimisation.
  d. Distribution Query Execution.
- The query decomposition phase is the initial stage of query processing, and its goals are to convert a high-level query into a relational algebra query and to validate that query for syntactic and semantic correctness.
- The exercise of save and processing data inside a certain geographical location is referred to as data localisation. It might be prompted by a number of considerations, including national rules or regulations, security and privacy concerns,or a desire to keep data closer to users.
- To reduce data transfer a cross sites, the global optimizer develops a distributed execution plan. The plan describes the sequence in which query stages must be performed, as well as the activities involved in transmitting intermediate results.

- All sites with query fragments execute the final layer. Each sub-query done at a single site is then optimized and executed sing the local schema of that site.

9.7Self-AssessmentQuestions

1. What is Query Processing?
2. What are the objectives of Query Processing?
3. Explain the various Characterisation of Query Processing.
4. What are the different layers of Query Processing?
5. What is Query decomposition?
6. What is the Localisation of distributed data?
7. What are the steps involved in Query Decomposition?
8. What is the goal of the Localization of distributed data?

9.8 References

- Atkinson, M.P. (1981). *Database*. Maidenhead: Pergamum Info Tech.
- Frank, L. and Helmersen, O. (1988). *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston, CT: Online, Inc.
- Kambayashi, Y. (1982). *Database*. Rockville, MD: Computer Science Press.

UNIT 10

DISTRIBUTED QUERY OPTIMISATION

Learning Outcomes

- Students will be capable of learning about the Factors governing query optimisation.
- Students will be able to understand Centralised query optimisation.
- Students will be able to understand Ordering of Fragment Queries.
- Students will be able to understand Distributed Query Optimization algorithms.

Structure

10.1 Factors governing query optimisation

Selecting the optimal execution plan for a given query within the restrictions of available resources is the aim of query optimization. The query describes the intended output, or the user's purpose, but it makes no mention of how the output should be generated. Query optimization is the process of making a database query function as efficiently as possible. Query optimization in database management systems refers to the process of figuring out the most effective way to run a SQL statement. The optimiser can combine, restructure, and process data in any sequence as SQL is a nonprocedural language. The optimization of DBMS queries is widely used in database design.

The following are the various principles of Query Optimisation:

1. Get to know how your database handles query execution. Query optimization begins with an understanding of the database's operation. For this, there are different commands in different databases. For example, use the "EXPLAIN [SQL Query]" keyword in MySQL to show the problem plan. Use the Oracle command "EXPLAIN PLAN FOR [SQL QUERY]" to view the query plan.
2. Get the least amount of data feasible. Because the database must expand to accommodate additional resources in order to analyze and retain these records, the more information that is returned from the query, the more. Avoid using 'SELECT *' if all you need to do is retrieve a single column from a table.
3. Keep intermediate results.The logic for a query might be rather complex at times. Subqueries, inline views, and UNION-style statements can be used to achieve the desired results. The transitional results for those techniques are not retained in the database, but are used directly within the query. This can cause problems, especially if the transitional results have a large number of rows.

Many factors influence query optimisation in a database management system (DBMS):

1. The number of I/O requests connected with a single file system access.
2. The amount of CPU works necessary to identify which rows satisfy the query criteria.
3. The resources needed to sort or group the data.

4. A variety of things can have an impact on query performance. The following data, cluster, and database actions all have an impact on how rapidly your queries are processed.
5. The number of compute nodes, processors, or slices - A compute node is divided into slices. More nodes imply more processors and more slices, allowing your queries to run quicker by running portions of the query concurrently across the slices. But, more nodes equal more expense, so you must find the right mix of cost and performance for your system. See Data warehouse system architecture for additional information on Amazon Redshift cluster architecture.
6. Node types -Amazon Redshift clusters can use either dense storage or dense computing nodes. Dense storage node types are recommended for large data storage requirements, whilst dense compute node types are intended for performance-intensive tasks.
7. Data Distribution: The way data is distributed across the database and indexed can have a substantial impact on query performance. A well-designed database with adequate indexes can boost query performance significantly.
8. Joins and Subqueries - Because joining numerous tables or utilising subqueries can be computationally expensive, it's crucial to keep the amount of joins and subqueries in a query to a minimum.
9. Selectivity and Cardinality - A query's selectivity and cardinality refer to how specific or selective it is, as well as the number of rows it is expected to return. Queries with a smaller cardinality will often run faster than those with a greater cardinality.
10. Server Resources - The amount of server resources available, such as RAM, CPU, and I/O, will affect query performance
11. Query Complexity - The query's complexity, such as the number of calculations and tables, will influence how much work the database has to do to execute the query.
12. Database Configuration - Database configuration, such as buffer pool settings, can have an impact on query performance.
13. Execution Plan - The execution plan is the sequence of steps that the DBMS takes to execute a query; the optimiser will generate various plans and select the best one based on specific criteria.
14. Statistics - The DBMS relies on statistics to determine how to execute a query, and the correctness of the information can influence the optimiser's conclusion.

You can help ensure that your database queries perform as effectively as possible by knowing and optimising all the above-mentioned factors.

10.2 Centralised Optimisation

In a centralised system, query processing is carried out with the following goals in mind:

1. Minimisation of query response time (the time it takes to deliver results for a user's query).
2. Increase system.
3. Decrease the processing memory and storage required.
4. Boost parallelism.

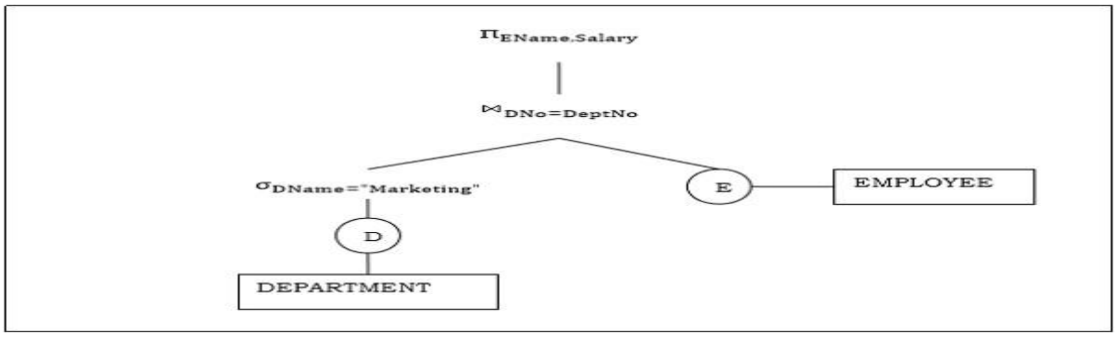


Fig 10.1 Query optimization in DBMS

1. **Query Parsing and Translation**

   First, the SQL query is examined. After that, it is parsed to verify the data type and look for syntactical errors. The query is broken up into more manageable query fragments if it passes this test. Each block is then converted into a corresponding relational algebra statement.
2. **Steps for Query Optimisation**

   Query optimisation consists of three steps: query tree generation, plan generation, and plan code generation.

Step 1 − Query Tree Generation

It is a tree data structure that represents a relational algebra statement. The query tables are represented as leaf nodes. Internal nodes represent relational algebra operations. The root reflects the query as a whole.When an internal node's operand tables are available, it is performed. The result table is then used to replace the node. This process is repeated for all internal nodes until the root node is performed, and the result table takes its place.

Step 2 − Query Plan Generation

This tree is formed, a query plan is created. An enlarged query tree with access paths for each action in the query tree is called a query plan. The relational operations of the tree are described by access paths. An access path that offers details on how to use the B+ tree index for selection, for instance, might be present in a selection procedure.A query plan also outlines the use of temporary tables, the pipelineing/combining of operations, and the transfer of intermediate tables from one operator to the next.

Step 3− Code Generation

The final phase in query optimisation is code generation. It is the query's executable form, which varies depending on the type of underlying operating system. The Execution Manager executes the query code and generates the results.

- **Approaches to Query Optimisation**
  - For query optimization, the most widely used approaches are heuristic-based algorithms and exhaustive search.

1. **Exhaustive Search Optimisation**

   These strategies produce all feasible query plans for a query and then select the best plan. Whilst these strategies yield the optimum solution, they have an exponential time and space complexity because to the huge solution space. For instance, consider the dynamic programming technique.
2. **Heuristic Based Optimisation**

   For query optimisation, heuristic-based optimisation employs rule-based optimisation methodologies. The time and space complexity of these algorithms is polynomial, as opposed to the complexity of search-based algorithms. These algorithms, however, do not always yield the optimum query strategy.

   Some examples of typical heuristic rules are given below as follows:

a. Before joining operations, perform choose and project operations. This is accomplished by relocating the choosed and project operations lower in the query tree. This decreases the amount of tuples that can be joined.
b. Execute the most restrictive select/project procedures first, followed by the others.
c. Cross-product operations should be avoided since they produce very large intermediate tables.

- **Knowledge Check 1**

  Fill in the Blanks

  1. The final phase in query optimisation is ____________.
  2. __________strategies produce all feasible query plans for a query and then select the best plan.
  3. ___________is very important in the centraliseddatabase, and it is more important in the distributed database as join between the fragments may increase the communication time.

- **Outcome-Based Activity 1**

Illustrate centralize Optimisation with Proper Diagram.

10.3 Ordering of Fragment Queries

Join order is very important in the centraliseddatabase, and it is more important in the distributed database as join between the fragments may increase the communication time.

Join ordering

- Distributed INGRES
- System R*

Semi join ordering

- SDD-1

There are many assumptions that are related withthe main issues.

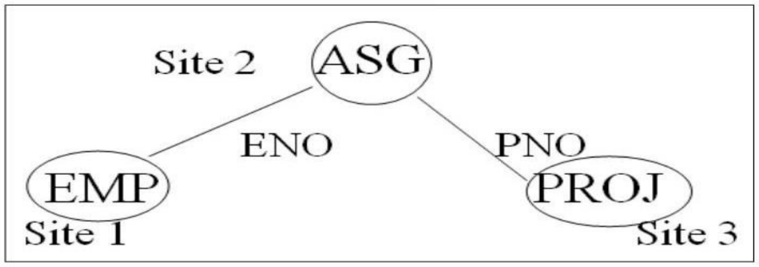
1. Relationships and fragments are indistinguishable.
2. The expense of local processing is removed.
3. Relations are transferred one set at a time.

4. The cost of transferring data to the result location to produce the final result is removed.

For order joins in the fragment queries, there are two fundamental methods.
1. Direct optimization of join ordering (as in the distributed INGRES method, for example)
2. To reduce the cost of communication, the joins are replaced by a collection of semi-joins.

Join Ordering - Example

Consider: **PROJ $\bowtie_{PNO}$ ASG $\bowtie_{ENO}$ EMP**

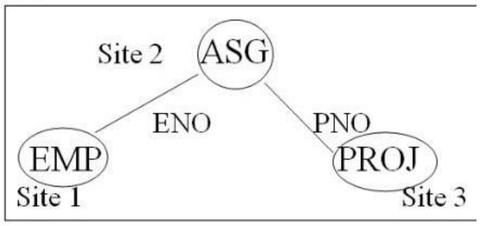


Join Ordering – Example (*cont.*)

PROJ $\bowtie_{PNO}$ ASG $\bowtie_{ENO}$ EMP

❖ Execution alternatives:

1. EMP → Site 2
   Site 2 computes EMP'=EMP⋈ASG
   EMP' → Site 3
   Site 3 computes EMP'⋈PROJ



2. ASG → Site 1
   Site 1 computes EMP'=EMP⋈ASG
   EMP' → Site 3
   Site 3 computes EMP'⋈PROJ

Join Ordering – Example (*cont.*)

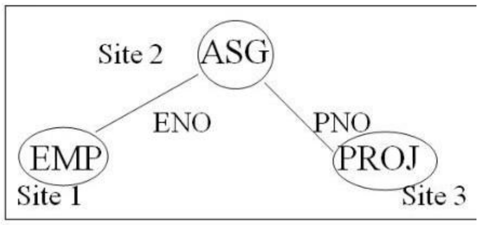
PROJ ⋈$_{PNO}$ ASG ⋈$_{ENO}$ EMP

3. ASG → Site 3

   Site 3 computes ASG'=ASG⋈PROJ

   ASG' → Site 1

   Site 1 computes ASG'⋈EMP



4. PROJ → Site 2

   Site 2 computes PROJ'=PROJ⋈ASG

   PROJ' → Site 1

   Site 1 computes PROJ' ⋈ EMP

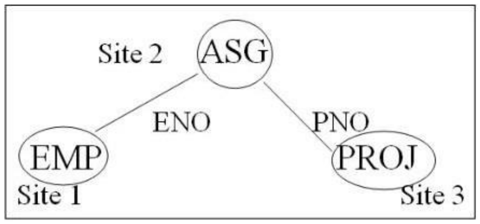
Join Ordering – Example (*cont.*)

PROJ ⋈$_{PNO}$ ASG ⋈$_{ENO}$ EMP

5. EMP → Site 2

   PROJ → Site 2

   Site 2 computes EMP⋈
PROJ⋈ASG

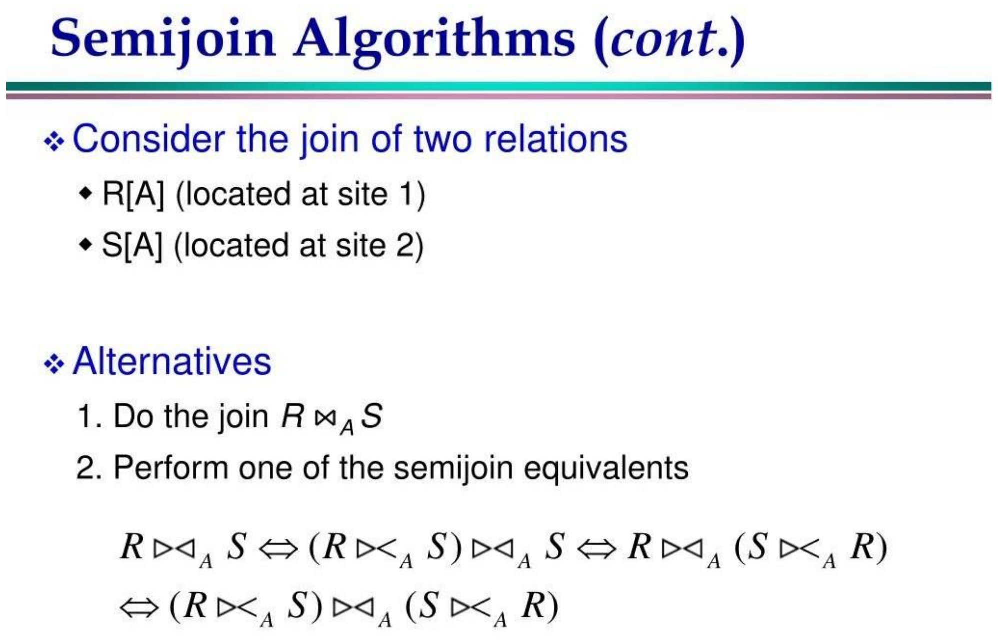


Semi join Algorithms

There are many shortcomings of the joining method that are mentioned below as follows:

- Transfer of the entire relations, which may also contain some of the useless tuples.
- Semi-join also reduces the size of the operand relations that needs to be transferred.

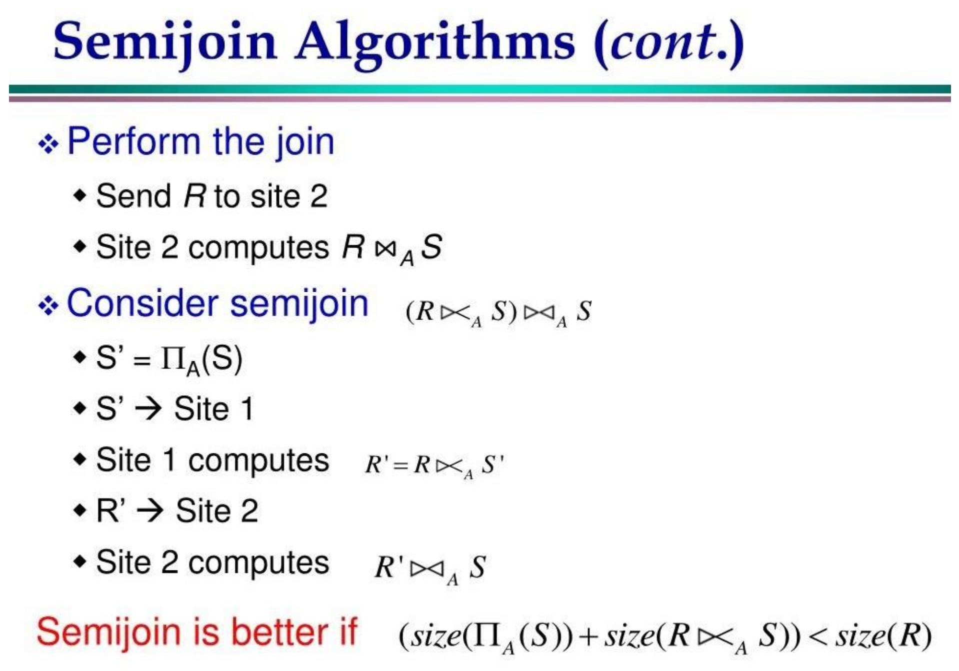
Semi join Algorithms is highly useful if the cost of developing and sending to the other site is less than the cost of sending the entire relation.



Semijoin Algorithms (*cont.*)

- Consider the join of two relations
  - R[A] (located at site 1)
  - S[A] (located at site 2)
- Alternatives
  1. Do the join $R \bowtie_A S$
  2. Perform one of the semijoin equivalents

$$R \bowtie_A S \Leftrightarrow (R \ltimes_A S) \bowtie_A S \Leftrightarrow R \bowtie_A (S \ltimes_A R)$$
$$\Leftrightarrow (R \ltimes_A S) \bowtie_A (S \ltimes_A R)$$



Semijoin Algorithms (*cont.*)

- Perform the join
  - Send R to site 2
  - Site 2 computes $R \bowtie_A S$
- Consider semijoin $(R \ltimes_A S) \bowtie_A S$
  - S' = Π_A(S)
  - S' → Site 1
  - Site 1 computes $R' = R \ltimes_A S'$
  - R' → Site 2
  - Site 2 computes $R' \bowtie_A S$

Semijoin is better if $(size(\Pi_A(S)) + size(R \ltimes_A S)) < size(R)$

10.4 Distributed Query Optimization Algorithms

It is the process of designing a strategy for a query's execution to a distributed database system. We refer to the plan as a query execution plan. Logical data units in a distributed database system are queries and schema. In a relational distributed relation database system, for instance, relations are logical units of data. These units could be dispersed at the most basic physical level. The fragments in the distributed system are allocated to several database servers and may be redundant or duplicated.

Distributed query optimisation necessitates the evaluation of a large number of query trees, each of which produces the required query results. This is primarily due to the high amount of replicated and fragmented data present. As a result, the goal is to identify an optimal solution rather than the best one.

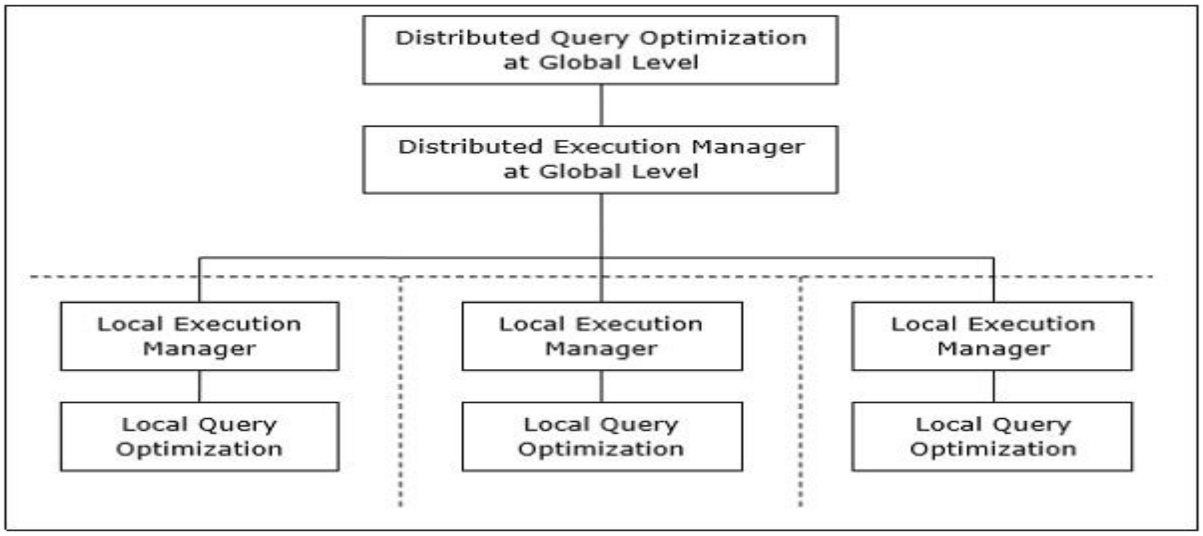


Fig 10.2 Distributed query optimization

The primary concerns for distributed query optimisation are as follows:

1. The most efficient use of distributed system resources.
2. Consider trading.
3. The query's solution space is reduced.

- **Optimum Resource Usage in a Distributed System**

A distributed system employs a number of database servers located across multiple sites to carry out query-related processes.

The methodologies for optimal resource usage are as follows:

1. **Operation Shipping**- Rather than at the client site, the process is performed at the location where the data is stored. After then, the client's website receives the results. For operations where both operands are present at the same location, this is helpful. Two instances are Select and Project operations.
2. **Data Shipping-** The database server receives the data pieces and executes the actions on them. This is applied to operations in which the operands are dispersed among several places. This is also suitable for systems with low connection costs and processors local to the client-server network that operate much more slowly.
3. **Hybrid Shipping-** This combines operation shipping with data. High-speed processors receive the data chunks and execute the operation there. After then, the client's website receives the results.

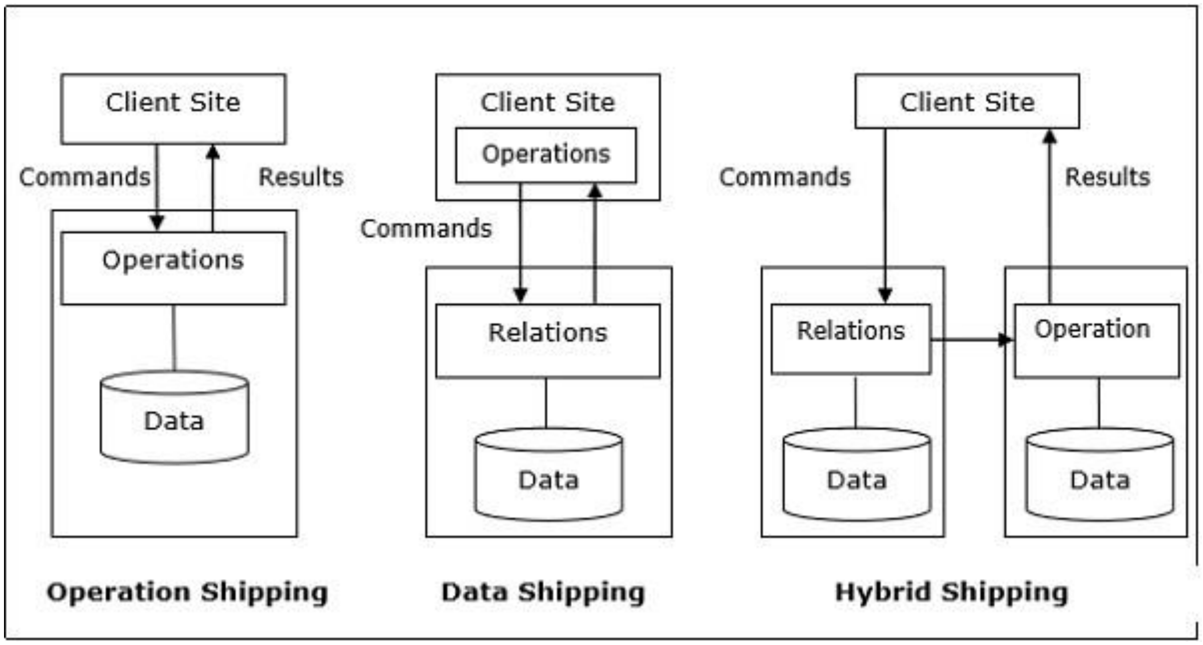


Fig 10.3 High-speed processors receives data

- **Knowledge Check 2**

  State True and False for the Following Sentences.

  1. Distributed query optimization is the process of formulating a strategy for a query's execution to a distributed database system.

2. In hybrid shipping, individual data pieces are sent to the database server, which handles the processes.
3. Join order is very important in the centralized database, and it is more important in the distributed database as join between the fragments may decreases the communication time.

- **Outcome-Based Activity 2**

List out some of the examples of Join and Semi-join algorithms**.**

10.5 Summary

- Query optimization is the process of figuring out how to run a SQL statement as efficiently as possible. The optimizer can combine, restructure, and process data in any sequence as SQL is a nonprocedural language.
- The process of query optimization involves three stages: the creation of query trees, plans, and plan codes.
- The following objective guides query processing in a centralized system: Shorten the time it takes to respond to a user's inquiry in terms of response time. Boost the system's throughput, or the quantity of requests handled in a predetermined length of time. Join ordering is very important in the centralised database, and it is more important in the distributed database as join between the fragments may increase the communication time.
- The distributed query optimization algorithm picks an optimal or satisfying plan by exploring areas of the combinatorial search space defined as the set of feasible query execution plans. The cost function to be optimised is referred to as the objective function.

10.6 Self-Assessment Questions

1. What is query optimisation?
2. What are the factors that governs query optimisation?
3. What is Centralised query optimisation?
4. What is the ordering of fragment queries?
5. What are the different approaches of query optimisation in a Centralised System?
6. What are the different steps of Query optimisation in a centralisedSystem?
7. Explain Query Tree generation.
8. Explain Distributed Query optimisation algorithms.

10.7 References

- Atkinson, M.P. (1981) *Database*. Maidenhead: Pergamum InfoTech.
- Frank, L. and Helmersen, O. (1988) *Database theory and practice*. Wokingham, England: Addison-Wesley Publishing Company.
- *Database users: 2nd Toronto conference: Selected papers* (1990). Canadian Association for Information Science.
- *Database* (no date). Weston Ct.: Online, Inc.
- Kambayashi, Y. (1982) *Database*. Rockville, MD: Computer Science Press.