1. **(a) Define distributed database system. What is the processing element in distributed computing system?**

**Answer:**

A distributed database System is a collection of databases which are distributed over different computers of a computer network.

Each site has autonomous processing capability and can perform local applications.

Each site also participates in the execution of at least one global application which requires accessing data at several sites.



**Distributed computing system** states that it is a number of autonomous processing elements (not necessarily homogeneous) that are interconnected by a computer network and that cooperate in performing their assigned tasks. The “processing element” referred to in this definition is a computing device that can execute a program on its own.

A processing element in a distributed computing system is a device or system that has its own processing capabilities and can execute tasks and communicate with other devices or systems in the network. A processing element can be a computer, a server, a workstation, a mobile device, or any other device that can perform computations and data processing. A processing element may also store and manage its own data, or access shared data from other devices or systems. A processing element is also known as a node in a distributed system.

**(b) Define each of the elements which is being distributed in distributed data processing environment.**

**Answer:**

* **Processing logic**: Processing logic in distributed data processing environment is the way of distributing the computation or processing of data across multiple nodes or machines that are connected by a network. This allows for faster and more efficient data processing, as well as scalability, availability, and fault-tolerance.
* **Function**: Various functions of a computer system could be delegated to various pieces of hardware or software.
* **Data**: Data used by a number of applications may be distributed to a number of processing sites.
* **Control**: The control of the execution of various tasks might be distributed instead of being performed by one computer system

**(c) Describe how improved performance can be achieved in distributed database system.**

**Answer:**

Improved performance can be achieved in distributed database system by using various techniques and strategies, such as:

1. A DDBMS fragments the conceptual database, enabling data to be stored in close proximity to its points of use (data localization).
   * Since each site handles only a portion of the database, contention for CPU and I/O services is not as severe as for centralized databases
   * Localization reduces remote access
2. The inherent parallelism of distributed systems may be exploited for inter-query and intra-query parallelism.
   * Inter-query parallelism results from the ability to execute multiple queries at the same time.
   * On the other hand, intra-query parallelism is achieved by breaking up a single query into a number of subqueries each of which is executed at a different site, accessing a different part of the distributed database.

(d) **Explain the alternative approaches to place the database and applications across different sites for distributed database design.**

**Answer:**

* To place the database and applications across different sites, there are two alternatives:

1. **partitioned** (or non-replicated) and
2. **replicated**.
   * **Partitioned scheme**: Database is divided into a number of disjoint partitions each of which is placed at a different site.
   * **Replicated scheme**: It can be fully replicated where the entire database is stored at each site, or partially replicated where each partition of the database is stored at more than one site but not at all the sites.

* Two fundamental design issues are **fragmentation**- the separation of the database into partitions called **fragments**, and **distribution**- the optimum distribution of the database.

1. **(a) Define the terms i) autonomy ii) distribution**

**Answer:**

i) **Autonomy**: It refers to the distribution of control, not of data. It indicates the degree to which individual DBMSs can operate independently. It is a function of a number of factors:

* whether the component systems exchange information
* whether they can independently execute transactions
* whether one is allowed to modify them.

Dimensions of autonomy:

* **Design autonomy**: Individual DBMSs are free to use the data models and transaction management techniques that they prefer.
* **Communication autonomy**: Each of the individual DBMSs is free to make its own decision as to what type of information it wants to provide to the other DBMSs or to the software that controls their global execution.
* **Execution autonomy**: Each DBMS can execute the transactions that are submitted to it in any way that it wants to.

The distribution dimension of the taxonomy deals with data. We consider the physical distribution of data over multiple sites, the user sees the data as one logical pool.

We consider two classes of DBMSs distribution:

1. client/server distribution
2. peer-to-peer distribution (full distribution).

The **client/server distribution** concentrates data management duties at servers while the clients focus on providing the application environment including the user interface.

In **peer-to-peer distribution**, there is no distinction of client machines versus servers. Each machine has full DBMS functionality and can communicate with other machines to execute queries and transactions.

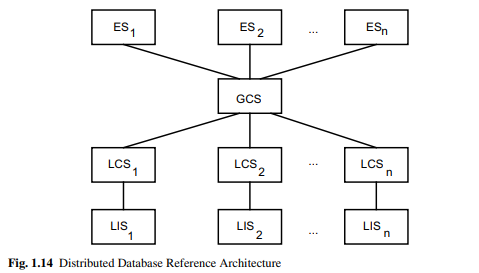
[N.B. -> A (**autonomy**), D (**distribution**), and H (**heterogeneity**). The alternatives along each dimension are identified by numbers 0, 1 or 2. Along the autonomy dimension, 0 represents tight integration, 1 represents semiautonomous systems and 2 represents total isolation. Along distribution, 0 is for no distribution, 1 is for client/server systems, and 2 is for peer-to-peer distribution. Along the heterogeneous dimension, 0 identifies homogeneous systems while 1 stands for heterogeneous systems]

**(b) What is the significance of (A0, D2, H1)**

**Answer:**

**(c) Describe distributed database reference architecture for peer-to-peer distributed database system.**

**Answer:**

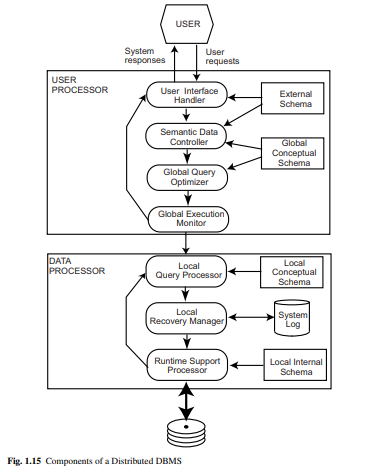


The physical data organization on each machine is different. There needs to be an individual internal schema definition at each site (local internal schema (LIS)), The enterprise view of the data is described by the global conceptual schema (GCS)- it describes the logical structure of the data at all the sites.

To handle fragmentation and replication, the logical organization of data at each site needs to be described. Therefore, there needs to be a third layer in the architecture, the local conceptual schema (LCS). The GCS is the union of the LCSs. User applications and user access to the database is supported by external schemas (ESs).

**(d) Show the components of the distributed DBMS using a diagram.**

**Answer:**



1. **(a) Describe top-down design strategies for designing distributed database.**

**Answer:**

The activity begins with a requirements analysis that defines the environment of the system and elicits both the data and processing needs of all potential database users.

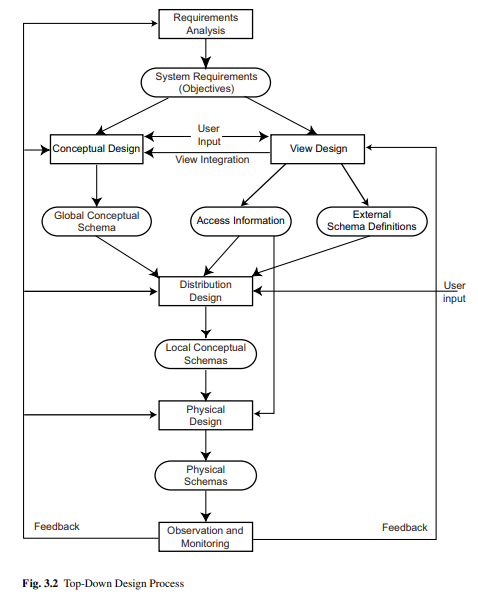
The requirements study also specifies where the final system is expected to stand with respect to the objectives of a distributed DBMS.

The requirements document is input to two parallel activities: view design (interfaces) and conceptual design (entity types and relationships).

The GCS and access pattern information collected by view design are inputs to the distribution design step. This step is used to design the local conceptual schemas by distributing the entities over the sites.

Rather than distributing relations, they are divided into sub-relations, called fragments, which are then distributed.

The last step in the design process is the physical design, which maps the local conceptual schemas to the physical storage devices available at the corresponding sites. The inputs to this process are the local conceptual schema and access pattern information about the fragments in these.



**(b) Write down the advantages of fragmentation.**

**Answer:**

Fragmentation in distributed database is the process of dividing a database into smaller parts or fragments that are stored on different computers or nodes within a network. Some of the advantages of fragmentation are:

* It enhances performance by allowing parallel processing of queries on different fragments.
* It improves data availability by providing multiple copies of data at different sites.
* It distributes the processing load by reducing the network traffic and the contention for resources.
* It allows local query optimization methods for some queries as the data is available locally.
* It maintains the security and privacy of the database system by restricting access to certain fragments.

**(c) Explain the use of reconstruction rule to ensure the correctness of fragmentation.**

**Answer:**

The following three rules together ensure that the database does not undergo semantic change during fragmentation:

1. ***Completeness***: If a relation instance **R** is decomposed into fragments R1, R2, …, Rn, each data item (HF – item refers tuple, VF – item refers attribute) that can be found in **R** can also be found in one or more of **Ri**’s (similar to lossless decomposition).
2. ***Reconstruction***: If a relation **R** is decomposed into fragments R1, R2, …, Rn, it should be possible to define a relational operator ∇ such that R = ∇Ri, ∀Ri ∈ FR. The reconstructability of the relation from its fragments ensures that constraints defined on the data in the form of dependencies are preserved.
3. ***Disjointness***: If a relation **R** is horizontally decomposed into fragments R1, R2, …, Rn and data item di is in Rj, it is not in any other fragment Rk (k ≠ j). In case of vertical partitioning, disjointness is defined only on the **nonprimary key** attributes of a relation.

**(d) Describe how primary horizontal fragmentation can be obtained using the PHORIZONTAL algorithm.**

**Answer:**

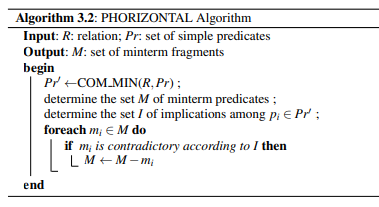
A primary horizontal fragmentation is defined by a selection operation on the owner relations of a database schema. For a given relation R, its horizontal fragments are given by

Ri = σ Fi (R), 1 ≤ I ≤ w

where Fi is the selection formula used to obtain fragment Ri. If Fi is in conjunctive normal form, it is a minterm predicate (mi).

Primary horizontal fragmentation is a technique of dividing a table into smaller sub-tables based on some simple conditions or predicates. The PHORIZONTAL algorithm is a method of obtaining primary horizontal fragmentation for a distributed database. The algorithm works as follows:

* **Input**: A relation R with attributes [A1, A2, …, An] and a set of simple predicates P = {P1, P2, …, Pm} that define the conditions for fragmentation.
* **Output**: A set of fragments F = {F1, F2, …, Fk} that are disjoint and complete, meaning that each tuple of R belongs to exactly one fragment and the union of all fragments is equal to R.



1. **(a) Explain the heuristic approaches for the vertical fragmentation of global relations.**

**Answer:**

A vertical fragmentation of a relation R produces fragments R1, R2, …, Rn, each of which contains a subset of R’s attributes as well as the primary key of R.

If a relation has *m* non-primary key attributes, the number of possible *fragments is equal to B(m), which is the mth Bell number (for large m, it is* mm).

**Two** types of **heuristic approaches** exist for the **vertical fragmentation** of global relations:

* **Grouping**: Starts by assigning each attribute to one fragment, and at each step, joins some of the fragments until some criteria is satisfied.
* **Splitting**: Starts with a relation and decides on beneficial partitioning based on the access behavior of applications to the attributes.

Splitting generates non-overlapping (non-primary key attributes) fragments whereas grouping typically results in overlapping fragments.

The replication of the global relation’s key in the fragments in vertical fragmentation allows the reconstruction of the global relation. Therefore, splitting is considered only for those attributes that do not participate in the primary key.

**(b) What is semantic integrity control? Define semantic integrity constraints.**

**Answer:**

* **Semantic integrity control** ensures database consistency by rejecting update programs which lead to inconsistent database states or by activating specific actions on the database states which compensate for the effect of update programs.
* **Semantic integrity constraints** are rules that represent the knowledge about the properties of an application. They define static or dynamic application properties which cannot be directly captured by the object and operation concepts of a data model**.**

A database state is said to be consistent if the database satisfies a set of constraints called **semantic integrity constraints**. Maintaining a consistent database requires various mechanisms such as **Concurrency control, Reliability, Protection and Semantic integrity control.**

**(c) Explain how views can provide data security. Create a view to see the employee details of a particular Job Title.**

**Answer:**

A view is a dynamic window in the sense that it reflects all updates to the database. An external schema can be defined as a set of views and/or base relations. Besides their use in external schemas, views are useful for ensuring data security in a simple way. By selecting a subset of the database, views hide some data. If users may only access the database through views, they cannot see or manipulate the hidden data, which are therefore secure.

CREATE VIEW SYSAN(ENO, ENAME)

AS SELECT ENO, ENAME

FROM EMP

WHERE TITLE = ‘’SYST. ANA’’

**(d) Define different aspects of data security.**

**Answer:**

Data security is an important function of a database system that protects data against unauthorized access. Data security includes two aspects:

* + - Data protection and
    - Authorization control

Data protection is required to prevent unauthorized users from understanding the physical contents of data.

The main data protection approach is Data encryption. Encrypted data can be decrypted (decoded) only by authorized users who know the code.

Authorization control guarantees that only the authorized users perform operations they are allowed to perform on the database.

Different users may have access to a large collection of data under the control of a single centralized or distributed system.

The centralized or distributed DBMS must be able to restrict the access of a subset of the database to a subset of the users.

Authorization control is provided by the operating system and more recently by distributed operating system.

1. **(a) What is integrity constraint? Specify the following constraints for the given database where the primary keys are underlined:**

***Emp*(eno, ename, title), *proj*(pno, pname, budget), *asg*(eno, pno, resp, dur)**

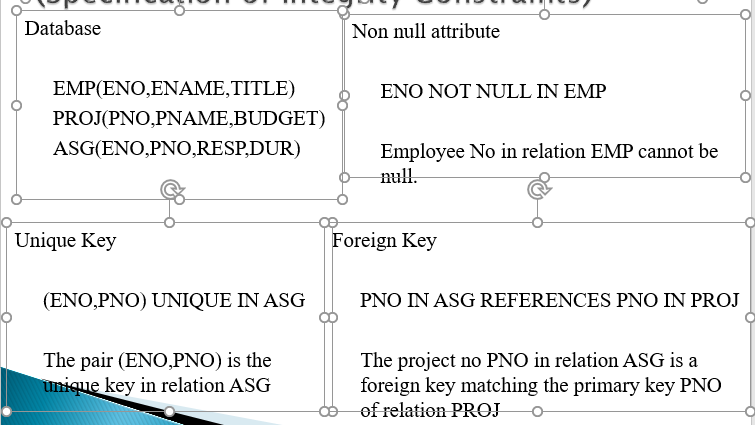
1. **The employee number functionally determines the employee name.**
2. **Only the tuples whose budget is 0 may be deleted.**
3. **The project no *pno* in relation *asg* is a foreign key matching the primary key *pno* of relation *proj*.**

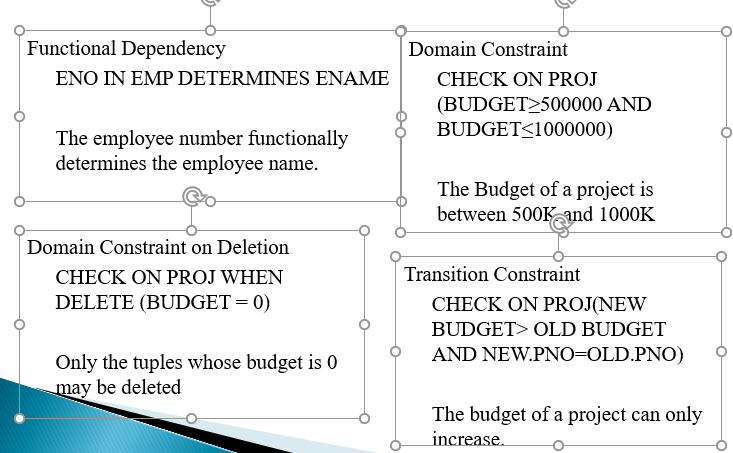
**Answer:**

A database state is said to be consistent if the database satisfies a set of constraints called semantic integrity constraints. Maintaining a consistent database requires various mechanisms such as **Concurrency control, Reliability, Protection and Semantic integrity control.**

**Semantic integrity control** ensures database consistency by rejecting update programs which lead to inconsistent database states or by activating specific actions on the database states which compensate for the effect of update programs. **Semantic integrity constraints** are rules that represent the knowledge about the properties of an application. They define static or dynamic application properties which cannot be directly captured by the object and operation concepts of a data model**.**

There are two main types of integrity. One is **structural constraints** and another is **behavioral constraints**.

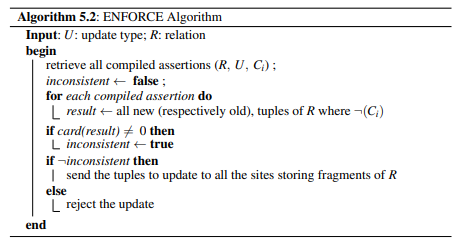




**(b) What do you mean by integrity enforcement? Describe the ENFORCE algorithm.**

**Answer:**

Enforcing Semantic integrity consists of rejecting update programs that violate some integrity constraints. A constraint is violated when it becomes false in the new database state produced by the update. A major difficulty in designing an integrity subsystem is funding efficient enforcement algorithm.



1. **(a) What is a query processor? Write down the characteristics of a query processor.**

**Answer:**

A query processor in a distributed database is a component that performs the functions of query decomposition, data localization, global query optimization, and distributed query execution. It takes a query on global data expressed in a high-level language (such as SQL) and transforms it into an optimized distributed query execution plan that minimizes the resource consumption (such as CPU, I/O, and communication costs) and maximizes the performance (such as response time and throughput) of the distributed system. A query processor in a distributed database has to deal with the challenges of data distribution, fragmentation, replication, and heterogeneity across multiple sites.

Important Characteristics of Query Processor:

* Languages
* Types of Optimization
* Optimization Timing
* Statistics
* Decision Sites
* Exploitation of the Network Topology
* Exploitation of Replicated Fragments
* Use of Semi-join

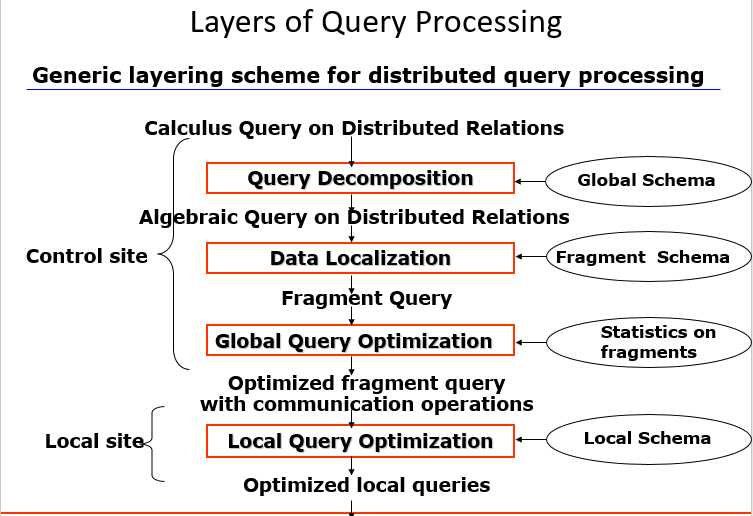
**(b) What are the main layers of query processing? Draw the layering scheme for distributed query processing.**

**Answer:**

Main layers of Query Processing

Query processing involves 4 main layers:

* Query Decomposition
* Data Localization
* Global Query Optimization
* Local Query Optimization



**(c) Write down the steps of query decomposition.**

**Answer:**

This is the first layer that decomposes the distributed calculus query into an equivalent algebraic query using global information.

Query decomposition can be viewed as four successive steps. The steps involved here are:

1. Calculus query is rewritten in a normalized form.
2. Normalized query is then analyzed semantically so that incorrect queries are detected and rejected as early a possible.
3. The correct query from the 2nd steps (which are still expressed in relational calculus) is simplified by eliminating redundant predicates.
4. Finally the corrected and simplified calculus query is restructured (converted) as an algebraic query.

**(d) Explain how local query optimization is achieved in distributed query processing.**

**Answer:**

Local query optimization is the process of finding the best way to execute a query on a single site in a distributed database system. It involves choosing the most efficient algorithm, data structure, and access method for each query operation, such as selection, projection, join, aggregation, etc. Local query optimization is achieved by using various techniques, such as:

**Cost-based optimization**: This technique estimates the cost of different query execution plans based on factors such as CPU time, disk I/O, network communication, and memory usage. The plan with the lowest cost is chosen as the optimal one. Cost-based optimization requires accurate statistics about the data distribution, size, and cardinality of the relations and their fragments.

1. **(a) Define normalization. Explain how an SQL query can be converted to normal forms with example.**

**Answer:**

Normalization is the process of organizing the data in the database to reduce redundancy and improve data integrity. Normalization is used to eliminate undesirable characteristics like insertion, update, and deletion anomalies, which can cause inconsistency and errors in the data. Normalization is also used to make the database more efficient and easier to query.

Normalization in a distributed database is similar to normalization in a centralized database, but it also considers the distribution of data across multiple sites.

**Find the names of employees who have been working on project P1 for 12 or 24 months.**

**The Query expressed in SQL is**

SELECT ENAME

FROM EMP, ASG

WHERE EMP.ENO = ASG.ENO

AND ASG.PNO = ‘P1’

AND DUR = 12 OR DUR = 24;

**The qualification in conjunctive normal form is**

EMP.ENO = ASG.ENO Λ ASG.PNO = “P1” Λ (DUR = 12 **∨** DUR = 24)

**While the qualification in disjunctive normal form is**

(EMP.ENO = ASG.ENO Λ ASG.PNO = “P1” Λ DUR = 12) **∨**

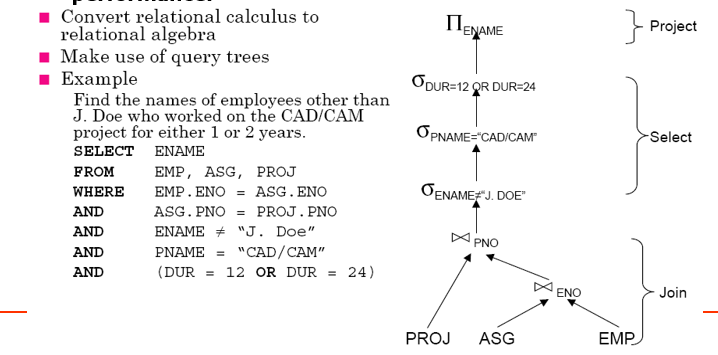
(EMP.ENO = ASG.ENO Λ ASG.PNO = “P1” Λ DUR = 24)

**(b) Describe how the rewriting step can be applied to a given query.**

**Answer:**

The last step of query decomposition rewrites the query in relational algebra. This is divided into two sub steps:

1. Straightforward transformation of the query from relational calculus into relational algebra.
2. Restructuring of the relational algebra query to improve performance.



**(c) What is multimedia database? Describe the organization of the multimedia data based on the principle of uniformity.**

**Answer:**

A **multimedia database** is a collection of related multimedia data that includes text, images, graphic objects, animation sequences, audio and video. A multimedia database management system (MMDBMS) is a framework that manages different types of multimedia data that can be stored, delivered and utilized in different ways. There are three classes of multimedia data: static media, dynamic media and dimensional media. Static media are time-independent, such as images and graphic objects. Dynamic media are time-dependent, such as audio, video and animation. Dimensional media are 3D data, such as games and computer-aided design programs. A multimedia database should support the integration, data independence, concurrency control, persistence, privacy, integrity control, recovery and query of multimedia data.

The principle of uniformity in multimedia data is the idea that all data in a database system must conform to some predefined structures and constraints, such as schemas, formats, vocabularies, etc. This principle ensures that the data can be easily interpreted, manipulated, and exchanged by different applications and users.

One way to organize the multimedia data based on the principle of uniformity is to use multidimensional data structures, such as k-d trees, quad-trees, or R-trees. These data structures can store and index spatial or temporal data in a hierarchical manner, allowing efficient retrieval and manipulation of multimedia objects such as images, videos, or audio files.

Another way to organize the multimedia data based on the principle of uniformity is to use compressed image representations, such as the discrete Fourier transform (DFT) or the discrete cosine transform (DCT). These representations can reduce the size and complexity of raw images by transforming them into frequency domains, where only the most significant coefficients are retained. This allows faster and more accurate similarity-based retrieval of images based on their features or content.

1. **(a) Define query optimization. What are the components of a query optimizer?**

**Answer:**

Query optimization in distributed database is the process of producing a plan for the processing of a query to a distributed database system. The plan is called a query execution plan. In a distributed database system, schema and queries refer to logical units of data, such as relations in a relational database. These units may be fragmented and replicated at the underlying physical level, and allocated to different database servers in the distributed system.

The query optimizer in a distributed database consists of three main components:

1. **The transformer**: This component transforms the query into an equivalent form that can be executed more efficiently. For example, it can apply algebraic rules, rewrite subqueries, or eliminate redundant operations.
2. **The estimator**: This component estimates the cost and cardinality of each possible query execution plan. The cost can be measured in terms of disk I/O, CPU usage, memory usage, or network traffic. The cardinality is the number of rows returned by each operation. The estimator uses statistics collected from the database and the network to make these estimates.
3. **The plan generator**: This component generates and compares different query execution plans based on the transformed query and the cost estimates. It chooses the plan with the lowest cost as the optimal plan. The plan specifies how to access, join, and aggregate data from different sites.

**(b) Define cost function with respect to total time and response time.**

**Answer:**

A cost function is a mathematical expression that measures how well a system or model performs in terms of its objectives. For example, in machine learning, a cost function can be used to evaluate how well a model fits the data and how complex the model is. A common goal is to minimize the cost function by adjusting the parameters of the model.

One possible way to define a cost function with respect to total time and response time is to use a weighted sum of both metrics. For example, if we have a system that processes request from users, we can define the cost function as:

C=w1​T+w2​R

where C is the cost, T is the total time, R is the response time, and w1​ and w2​ are positive weights that reflect the relative importance of each metric. The total time is the sum of the service time and the wait time for each request, while the response time is the elapsed time from when a user submits a request until they receive a response. A lower cost indicates a better performance of the system.

**(c) Define the terms for INGRES algorithm: i) substitution ii) detachment**

**Answer:**

The INGRES algorithm is a technique for query optimization in relational database systems. It decomposes a complex query into simpler subqueries that can be executed more efficiently. The terms for the INGRES algorithm are:

**Substitution**: This is the process of replacing the value of each tuple (row) in a subquery with the actual values from the corresponding relation (table) and simplifying the expression. For example, if we have a subquery q1 = select std\_name from std\_info where std\_id = 1, and a relation std\_info with the tuple (1, Ahmed, Information, 75), then substitution will replace std\_id = 1 with std\_id = (1, Ahmed, Information, 75) and simplify it to true. The result of q1 after substitution will be std\_name = Ahmed.

**Detachment**: This is the process of breaking down a query into two subqueries that have a common variable, which is the result of the first subquery. The second subquery uses the result of the first subquery as an input. For example, if we have a query q = select std\_name, std\_dpt from std\_info, std\_absence where std\_info.std\_id = std\_absence.std\_id and std\_absence.abs\_hrs > 1 and std\_absence.date = '2020-12-06', then detachment will decompose it into two subqueries: q' = select std\_id from std\_absence where abs\_hrs > 1 and date = '2020-12-06' and q" = select std\_name, std\_dpt from std\_info where std\_id in q'. The result of q" will be the final answer of q.

**(d) Define data mining for distributed database environment.**

**Answer:**

Data mining for distributed database environment is the process of extracting useful information and patterns from distributed data sources that are stored in different locations and may have different schemas and software. Data mining for distributed database environment aims to overcome the challenges of data heterogeneity, data security, communication costs, and scalability issues that arise in distributed database systems. Data mining for distributed database environment can use various techniques and algorithms to perform data analysis and knowledge discovery in a parallel, cooperative, or federated manner. Some of the common processes in data mining for distributed database environment are data preprocessing, data integration, data partitioning, data mining, and result integration. Some of the common algorithms in data mining for distributed database environment are association rule mining, classification, clustering, frequent itemset mining, and outlier detection. Data mining for distributed database environment can provide many benefits such as improved performance, reduced communication costs, enhanced data quality, increased scalability, and preserved data privacy.