Lab 13

1. Compare and contrast a DDBMS with a centralized DBMS. Under what circumstances would you choose a DDBMS over a centralized DBMS?

   Ans:-

Centralized DBMS	DDBMS
Data are store at specific location	Data is store where they are mostly relevant
Data security is difficult to manage	Desirable to enforce security to data
If failure happens we loss everything	Replication of data at different locations are a safety to avoid loss of data in case of h/w or s/w failures
We need large storage space	Storage space depends on the necessity of the data for the site operations
Does not reflect organizational structure	Reflects organizational structure
Commonly used design is applied	Needs expertise and experience to design and maintain
Cheap	Expensive

If I am developing an enterprise architecture application I want to make sure all my independent part of my applications gets tailored access to the database components. The more the distributed the system is the easier to maintain and fix the application which in a long run reduces future software maintenance cost. The DDBMS goes with future advancement of technology, networking principle and accessibility far better than centralized DBMS.

2. What is the difference between a homogeneous and heterogeneous DDBMS? Under what circumstances would such systems generally arise?

   Ans:-

In a homogeneous DBMS, all sites in the database network use the same DBMS product, which simplifies design and implementation of the database system. Such a system facilitates incremental growth and performance improvement.

In contrast, in a heterogeneous DBMS, each site may use different DBMS products, data models and hardware products. As a results, gateways or interfaces are required to integrate the different DBMS and hardware products in the network.

Heterogeneous DBMS arise when existing independent database sites using different DBMS products are integrated into a single database system (distributed). For example, when an existing database system of one company is acquired by another company having a different DBMS product.

Homogeneous uses same DDBMS product, easier to design and mange and provides incremental growth and allows increased performance.

Heterogeneous DDBMS : uses different DBMS products with

3. Describe alternative schemes for fragmenting a global relation. State how you would check for correctness to ensure that the database does not undergo semantic change during fragmentation.
   Ans:-

A global relation (table) in distributed database system can be fragmented in three ways:

Horizontal fragmentation – consists of subsets of tuples in the relation

Vertical fragmentation – consists of subsets of attributes in the relation

Hybrid (Mixed) fragmentation – consists of horizontal fragments that are vertically fragmented or vertical fragments that are horizontally fragmented.

Completeness – to ensure no loss of data, each data item in a relation must appear at least in one fragment.

Reconstruction – to maintain functional dependencies after fragmentation, a relational operation must be defined to reconstruct the relation from its fragments.

Disjointness – a data item in one fragment should not appear in another fragment, except for the primary key attributes in vertical fragmentation. In vertical fragmentation, the primary key attributes must be repeated in each fragment to maintain integrity (for reconstruction).

The following rules are applied to ensure correctness of database fragmentation

4. Which kind of transparency is applicable only to heterogeneous DDBMSs? Why?
   Ans:-

   Because, different sites may use different DBMS products. The DDBMS should ensure that users are not aware of the difference between the local DBMS product and other products in the network.

5. A DDBMS must ensure that no two sites create a database object with the same name. One solution to this problem is to create a central name server. What are the disadvantages with this approach? Propose an alternative approach that overcomes these disadvantages.

   Ans:-

Having a central naming server can have the following disadvantages:

Branch or site DBMSs could be deprived of some local autonomy.

The central server can be a bottleneck to the system, resulting in performance degradation.

Failure of the central server can lead to low availability of the database as the branch sites cannot create new database objects.

Alternative approach: prefix each database object with an identifier of the creating site.