Assignment 3
Of
ADBMS

Submitted By:-
Aman Chauhan
D3 CSE A1
1805158

Q1. Explain about Transparencies in a DDBMS.
Ans. Distribution transparency is the property of distributed databases by the virtue of which the internal details of the distribution are hidden from the users. The DDBMS designer may choose to fragment tables, replicate the fragments and store them at different sites. However, since users are oblivious of these details, they find the distributed database easy to use like any centralized database.

The three dimensions of distribution transparency are −

- Location transparency
- Fragmentation transparency
- Replication transparency

Q2. Differentiate between Parallel Databases and Distributed Databases.

Ans.

Distributed Databases	Parallel Databases

i) A distributed database is a database in which all the storage devices are not connected to a common processor	i) A parallel database is a database that helps to improve the performance by parallelizing various operations such as data loading, building indexes, and evaluating queries.
ii) Nodes of a distributed database are in different locations.	ii) Nodes of a parallel database are in the same location.
Iii) A distributed database is slower than a parallel database.	iii) A parallel database is faster than a distributed database.
iv) The overhead of a distributed database is higher	iv) The overhead of a parallel database is lower.

Q3. Explain Date's Twelve Rules for a DDBMS.

Ans. 1) Local autonomy:-

The sites in a distributed system should be *autonomous*. In this context, autonomy means that:

- local data is locally owned and managed;

- local operations remain purely local;
- all operations at a given site are controlled by that site.

2) No reliance on a central site:-

There should be no one site without which the system cannot operate. This implies that there should be no central servers for services such as transaction management, deadlock detection, query optimization, and management of the global system catalog.

3) Continuous operation:-

Ideally, there should never be a need for a planned system shutdown, for operations such as: n adding or removing a site from the system; n the dynamic creation and deletion of fragments at one or more sites.

4) Location independence:-

Location independence is equivalent to location transparency. The user should be able to access the database from any site. Furthermore, the user should be able to access all data as if it were stored at the user’s site, no matter where it is physically stored.

5) Fragmentation independence:-

The user should be able to access the data, no matter how it is fragmented.

6) Replication independence:-

The user should be unaware that data has been replicated. Thus, the user should not be able to access a particular copy of a data item directly, nor should the user have to specifically update all copies of a data item.

7) Distributed query processing:-

The system should be capable of processing queries that reference data at more than one site.

8) Distributed transaction processing:-

The system should support the transaction as the unit of recovery. The system should ensure that both global and local transactions conform to the ACID rules for transactions, namely: atomicity, consistency, isolation, and durability.

9) Hardware independence:-

It should be possible to run the DDBMS on a variety of hardware platforms.

10) Operating system independence:-

As a corollary to the previous rule, it should be possible to run the DDBMS on a variety of operating systems.

11) Network independence:-

Again, it should be possible to run the DDBMS on a variety of disparate communication networks.

12) Database independence:-

It should be possible to have a DDBMS made up of different local DBMSs, perhaps supporting different underlying data models. In other words, the system should support heterogeneity. The last four rules are ideals. As the rules are so general, and as there is a lack of stand- ards in computer and network architectures, we can expect only partial compliance from vendors in the foreseeable future.

Q4. Discuss how data marts differ from data warehouses and identify the main reasons for implementing a data mart.

Ans.

Data Warehouses	Data Mart
i) Data Warehouse is a large repository of data collected from different sources.	i) Data Mart is only a subtype of a data warehouse.

ii) Data Warehouse is focused on all departments in an organization .	ii) Data Mart focuses on a specific group.
Iii) Data Warehouse designing process is complicated	Iii) The Data Mart process is easy to design.
iv) Data Warehouse takes a long time for data handling	iv) Data Mart takes a short time for data handling.
v) Data Warehouse size range is 100 GB to 1 TB+	v) Data Mart size is less than 100 GB.
vi) Data Warehouse implementation process takes 1 month to 1 year	vi) Data Mart takes a few months to complete the implementation process.

The main reasons for implementing a data mart are:-

- Efficient access — A data mart is a time-saving solution for accessing a specific set of data for business intelligence..
- Inexpensive data warehouse alternative — Data marts can be an inexpensive alternative to developing an enterprise data warehouse,

where required data sets are smaller. An independent data mart can be up and running in a week or less.
- Improve data warehouse performance — Dependent and hybrid data marts can improve the performance of a data warehouse by taking on the burden of processing, to meet the needs of the analyst. When dependent data marts are placed in a separate processing facility, they significantly reduce analytics processing costs as well.
- Data maintenance — Different departments can own and control their data.
- Simple setup — The simple design requires less technical skill to set up.
- Analytics — Key performance indicators (KPIs) can be easily tracked.
- Easy entry — Data marts can be the building blocks of a future enterprise data warehouse project.

Q5. Describe how star and snowflake schemas differ.

Ans.

Star Schema	Snowflake Schema
i) In the star schema, The fact tables and the dimension tables are contained.	i) In snowflake schema, The fact tables, dimension tables as well as sub dimension tables are contained.
ii) Star schema is a top-down model.	ii) It is a bottom-up model.
iii) It takes less time for the execution of queries.	Iii) It takes more time than star schema for the execution of queries.
iv) In star schema, Normalization is not used.	iv) Both normalization and denormalization are used.
v) It has fewer foreign keys.	v) It has more number of foreign keys.
vi) It has high data redundancy.	vi) It has low data redundancy.

Q6. Describe Codd's rules for OLAP.

Ans. Codd's rule for OLAP are:-

<u>1) Multidimensional conceptual view:-</u>OLAP tools should allow users with a multi-dimensional model

that keeps up a correspondence to users' views of the enterprise and is intuitively analytical and straightforward to use. Interestingly, this rule is given various levels of support by sellers who disagree that a multi-dimensional conceptual view of data can be delivered without multi-dimensional storage.

2) Transparency:-The OLAP technology has the underlying database and architecture, and the likely heterogeneity of input data sources that should be apparent to users. This necessity is to preserve the user's productivity and proficiency with familiar front-end environments and tools.

3) Accessibility:- The OLAP tool also allows access to data needed for the analysis from all heterogeneous enterprise data sources such as relational, non-relational, and legacy methods.

4) Consistent reporting performance:- With the number of dimensions, levels of aggregations, and the size of the database raises, users ought not to perceive any significant fall in performance. There should be no change in the way the key figures are calculated, and the system models must have

to be strong enough to cope with changes to the enterprise model.
5) Client-server architecture:- The OLAP system should be proficient enough to operate efficiently in a client-server environment. The architecture should permit optimal performance, flexibility, adaptability, scalability, and interoperability.

6) Generic dimensionality:- Every data dimension must be the same in both structure and operational capabilities, i.e., the basic structure, formulae, and reporting should not be biased towards any one dimension.

7) Dynamic sparse matrix handling:- The OLAP system should be able to cope up with the physical schema to the specific analytical model that optimizes sparse matrix handling to achieve and maintain the required level of performance.

8) Multi-user support:- The OLAP system should be able to hold up a group of users working at the same time on the same or different models of the enterprise's data.

9) Unrestricted cross-dimensional operations:- The OLAP system must be able to identify the dimensional hierarchies and automatically perform associated roll-up calculations across dimensions.

10) Intuitive data manipulation:- Slicing and cubing, consolidation (roll-up), and other manipulations can be accomplished via direct 'point-and-click' or 'drag-and-drop' actions on the cells of the cube.

11) Flexible reporting:- The capability of arranging rows, columns, and cells in a way that facilitates analysis by an intuitive visual presentation of analytical reports must exist.

12) Unlimited dimensions and aggregation levels:- Depending on business needs, an analytical model may have some dimensions, each having multiple hierarchies.