Denormalization

Denormalization is a database optimization technique to in which we add redundant data to tables, so we can query quickly. Because the join operation is computation and time consuming, so we add some tables together to create a big table, in this case, we can avoid costly joins in a relational database.

Normally, we would combine normalization and denormalization together to build our database.

Pros of Denormalization:-

1. Querying data is faster and more simple since we do fewer joins

Cons of Denormalization:-

1. Updates and inserts are more expensive.
2. Denormalization can make update and insert code harder to write.
3. Data may be inconsistent .
4. Data redundancy use more storage.

Shading

In DBMS, Sharding is partitioning a large DataBase into smaller database,

The most important reason to use shading are two. Firstly, we would be able to store the extremely large modern database. Secondly, Shading helps us to query the data much quicker

Other things about Sharding: Sharding makes the Database much more easily manageable. Sharding can be a complex operation sometimes

Some common Sharding or Data Partitioning Schemes

Horizontal or Range Based Sharding: In this case, the data is split based on the ***value ranges*** that are inherent in each entity.

Vertical Sharding: In this case, different features of an entity will be placed in different shards on different machines.

Key or hash based sharding: In this case, an entity has a value (Eg. IP address of a client application) which can be used as an input to a hash function and a resultant hash value generated. This hash value determines which database server(shard) to use.

Directory based sharding: Directory based shard partitioning involves placing a ***lookup service*** in front of the sharded databases.

Replica

Data Replication is the process of storing data in many sites or nodes. It is useful in improving the availability and reliability of data. It is simply copying data from a database from one server to another server.

Types of Data Replication –

1. **Transactional Replication: consistency is guaranteed.** Transactional replication is typically used in server-to-server environments.
2. **Snapshot Replication –** Snapshot replication is generally used when data changes are infrequent.
3. **Merge Replication –** Data from two or more databases is combined into a single database. Merge replication is the most complex type of replication because it allows both publisher and subscriber to independently make changes to the database. Merge replication is typically used in server-to-client environments. It allows changes to be sent from one publisher to multiple subscribers.

Replication Schemes –

1. Full Replication – Replication of the whole database at every site in the distributed system. This will improve the availability of the system.

Advantages of full replication:

- High Availability of Data.
- Faster execution of Queries. Improves the performance for retrieval of global queries as the result can be obtained locally from any of the local site.

Disadvantages of full replication:

- Concurrency is difficult to achieve in full replication.
- Slow update process

3. Partial Replication –Only partial of the database may be replicated.

Advantages of Partial replication

The number of copies of the fragment depends upon the importance of data.

ADVANTAGES OF DATA REPLICATION

- To provide a consistent copy of data across all the database nodes.
- To increase the availability of data.
- The reliability of data is increased through data replication.
- To perform faster execution of queries.

DISADVANTAGES OF DATA REPLICATION –

- More storage space is needed as storing the replicas of same data at different sites consumes more space.
- Data Replication becomes expensive when the replicas at all different sites need to be updated.
- Maintaining Data consistency at all different sites involves complex measures.