Assignment (Video- 19 to 21): Database System Impl. (COP6726)

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

1) **Properties of Hashing Functions (Deterministic and Look random):** a hash function is a one-way function on which Hashing algorithms are based, that means the output of the hash function is not feasible to be reversed or inverted. The hash function is required to be "deterministic" and "look Random" in nature. Deterministic means that the hash created for each input should be unique, the hash function should not generate similar hashes for different inputs. Every change to a message, even the smallest one, should change the hash value. It should be completely different. Diffusion means that changing a single character of the input will change many characters of the output. This property is needed so that hash function output looks random and its analysis becomes harder. Hence it provides better encryption security.

2) **Balls and Bins problem:** In probability theory the balls and bins problem are very popular problem because it has various applications in computer science. This problem actually incorporates M balls and N bins. there are multiple passes and in each pass one ball is put into a bin. Once all the balls are kept in bins. Then it is analyzed that how many balls are actually there in each bin. This is called load on a single bin. the problem asks the load on a single bin. The problem becomes unique because in each pass the ball could be kept in the bin differently like in uniform manner or non-uniform random manner a bin could be chosen to keep the ball. "power of two random choices" is a powerful balls-into-bins paradigm where each ball chooses two (or more) random bins and is placed in the lesser-loaded bin. This paradigm has found wide practical applications in shared-memory emulations, efficient hashing schemes, randomized load balancing of tasks on servers, and routing of packets within parallel networks and data centers.

3) **Iterator Model in Databases:** Each database operator (relational algebra) implements a common interface. The open function Reset internal state and prepare to deliver first result tuple. The next () function Deliver next result tuple or indicate EOF. and close () function Release internal data structures, locks, etc. Evaluation is driven by the top-most operator which receives open (), next (), next (), calls and propagates. This iterator model is also called Volcano model.

4) **Massively Parallel processing System:** Massively Parallel Processing systems are the database systems which are optimized for analytical workloads. These analytical tasks include aggregation and processing of large data sets. Unlike traditional database which saves each row in a table as an object the MPP databases save each column as an object. This mechanism allows MPP databases to process query very quickly and efficiently. Analytical databases distribute the workload across various machines these machines act as

processing nodes and participate parallelly to process the queries. These nodes all contain their own storage and compute capabilities, enabling each to execute a portion of the query.

5) **Partition Data and Partition Computation:** In many large-scale distributed systems data is also partitioned or divided so that the segregated data could be manages and accessed separately. This helps in reducing contention, improve scalability and optimize performance. Data Partitioning improves security and also provides operational flexibility as well as availability.

6) **Horizontal Vs Vertical Partitioning:** Horizontal partitioning is also called sharding in distributed processing. In this technique, each partition act as a separate data store, but the each and every single node or partition has the same schema to store data. each partition contains a specific subset of the data. On the other hand, we have vertical partitioning in this technique the fields are divided according to their pattern of use. more frequently accessed fields are placed in in one single vertical partition while less frequently used fields are kept in another.

7) **Hashing Partitioning:** As opposed to horizontal or vertical partitioning where data is put into different nodes in form of groups. In case of Hash partitioning the data is put into data store nodes in randomized form. Though there are no performance benefits associated with this type of partitioning techniques because it shuffles data across the table space randomly. The partitioning system can be used to efficiently match queries. It makes use of hashing algorithms to distribute the data across the device to space out the load. By this method, the partitions are approximately the same size. The data that can be partitioned is not historical in nature, and thus this method is very easy to use.

8) **Star vs Snowflake Schema:** A star schema is mainly used in data warehousing architectures. In this type of schema, a fact table references various dimension tables. such connectivity when viewed in a diagram looks like a star hence it is called a star schema. On the other hand, snowflake schema is same as star schema, the only difference is in the dimensions themselves. In a star schema each logical dimension is denormalized into one table, while in a snowflake, at least some of the dimensions are normalized.

9) **Load Balancing:** Database load balancing is a mechanism to distribute the workload among various database nodes evenly so that not one single computing resource or node gets overwhelmed by requests. A database load balancer is a middle ware between the application and database nodes. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.