Assignment (Video- 25 to 27): Database System Impl. (COP6726)

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

- 1) MonetDB: MonetDB is a column-oriented database management system that means it stores data in columns. The main characteristics of the Monet DB is that in case of large databases it provides high performance on complex queries. These complex queries may involve operations on hundreds of columns or large number of rows at a time. It is used in various applications where high performance of databases is required such as data mining, geographic information system, resource description framework etc. MondetDB is in memory optimized and it can also support json based documents. It has a scalable architecture. Vertical data organization and in-memory optimization guarantee fast query responses for very large datasets, without substantial hardware investments. in Monet DB the index management in automated hence no DBA work is required.
- 2) C_Store: C_Store is a database management system which stores data in columns instead of rows as in traditional database management systems. This database system is optimized for reading rather than writing. In various cases when large queries are operated on data in the result set only few columns are required. When the query is executed in a row bases relational database system then a large number of rows or records are retrieved and then required columns are extracted from the result set, in case of column based database the records themselves are saved in columns of rows not rows, hence when these complex queries are executed in C_Store database models the query execution is really fast and efficient. When we have a large dataset, with many rows and columns, only retrieving the columns that we need to execute our analytics query is substantially more efficient than retrieving all the records in the database.
- 3) Column Stores and Analytics: A column-based database is always preferred for analytics-based applications. This database system is optimized for reading. In case of analytics most of the queries are complex and heavy, in case of column-based database the records themselves are saved in columns not rows, hence when these complex queries are executed in column-based database models the query execution is really fast and efficient. When we have a large dataset, with many rows and columns, only retrieving the columns that we need to execute our analytics query is substantially more efficient than retrieving all the records in the database.
- 4) Column Store Vectorization: In The columnar database systems the data is stored in the form of columns, these columns are called vectors, the number of vectors in a columnar database is equal to number of column records. Vectorized code makes efficient utilization of CPU cache. Vectorized processing is done on columns,

and it defers the materialization of values from multiple columns into tuples (or rows) till very late in the query plan up till the result set is actually projected back to the user. In these cases, the query execution algorithms are rewritten for column-based processing. This makes sense because if the data is stored in the form of columns and processing code is optimized for row oriented database then after the columns are read the values from different columns to be stitched together to form a tuple then feed the tuple into a query operator that does conventional row-by-row processing. Forming tuples sooner during execution prevents running highly optimized query processing logic on columnar data.

5) Column Store Compression: In case of columnar databases column compression means that in order to compress the actual data which is present in columns some methods are used to accomplish that, the columnar format also allows us to use CPU-efficient compression schemes that are extremely lightweight and actually favor the performance of query processing rather than actually favoring the compression ratio, which can hurt your CPU efficiency tremendously. For example, "tokenization" is the most common type of compression where the actual repeated data values in the columns are replaced by a specific token that way data store could be optimized for memory. There are various other types of compression algorithm in a columnar database such as Gzip and Fast compression.