Roll No.: **74**	Name: **Akash Narendra Kinage**
Class: **SE-B**	Batch: **B4**

Q.1) Explain Database Normalization process.

ANS:
Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

First normal form (1NF)
Tables in 1NF must adhere to some rules:

- Each cell must contain only a single (atomic) value.
- Every column in the table must be uniquely named.
- All values in a column must pertain to the same domain.

Second normal form (2NF)

- Tables in 2NF must be in 1NF and not have any partial dependency (e.g. every non-prime attribute must be dependent on the table's primary key).

Third normal form (3NF)

- Tables in 3NF must be in 2NF and have no transitive functional dependencies on the primary key.

Boyce-Codd Normal Form (BCNF)

- A higher version of the 3NF, the Boyce-Codd Normal Form is used to address the anomalies which might result if one more than one candidate key exists. Also known as 3.5 Normal Form, the BCNF must be in 3NF and in all functional dependencies ($X \rightarrow Y$), X should be a super key.

Fourth Normal Form (4NF)

- For a table to in 4NF, it must be in BCNF and not have a multi-valued dependency.

Q.2) Explain Time stamp based protocol.

ANS:
Timestamp based protocol:

- The Timestamp-based protocol ensures that every conflicting read and write operations are executed in a timestamp order.
- This is the most commonly used concurrency protocol.
- The older transaction is always given priority in this method.
- It uses system time to determine the time stamp of the transaction

Advantages:

- Schedules are serializable just like 2PL protocols
- No waiting for the transaction, which eliminates the possibility of deadlocks!

Disadvantages:

- Starvation is possible if the same transaction is restarted and continually aborted.

Timestamp Ordering Protocol

- The timestamp-ordering protocol ensures serializability among transactions in their conflicting read and write operations. This is the responsibility of the protocol system that the conflicting pair of tasks should be executed according to the timestamp values of the transactions.
- The timestamp of transaction T_i is denoted as $TS(T_i)$.
- Read time-stamp of data-item X is denoted by R-timestamp(X).
- Write time-stamp of data-item X is denoted by W-timestamp(X).

Thomas' Write Rule

- This rule states if TS(Ti) < W-timestamp(X), then the operation is rejected and T_i is rolled back.
- Time-stamp ordering rules can be modified to make the schedule view serializable.
- Instead of making T_i rolled back, the 'write' operation itself is ignored.

Q.3) Explain Checkpoint in recovery.

ANS:

Checkpoint-Recovery is a common technique for imbuing a program or system with fault tolerant qualities, and grew from the ideas used in systems which employ transaction processing. It allows systems to recover after some fault interrupts the system, and causes the task to fail, or be aborted in some way. While many systems employ the technique to minimize lost processing time, it can be used more broadly to tolerate and recover from faults in a critical application or task.

The basic idea behind checkpoint-recover is the saving and restoration of system state. By saving the current state of the system periodically or before critical code sections, it provides the baseline information needed for the restoration of lost state in the event of a system failure. While the cost of checkpoint-recovery can be high, by using techniques like memory exclusion, and by designing a system to have as small a critical state as possible may minimize the cost of checkpointing enough to be useful in even cost sensitive embedded applications.

The checkpoint is used to declare a point before which the DBMS was in the consistent state, and all transactions were committed. During transaction execution, such checkpoints are traced. After execution, transaction log files will be created.
Upon reaching the savepoint/checkpoint, the log file is destroyed by saving its update to the database. Then a new log is created with upcoming execution operations of the transaction and it will be updated until the next checkpoint and the process continues.

Steps how to use Checkpoints in database:

- Write begin_checkpoint record into log.
- Collect checkpoint data in the stable storage.
- Write end_checkpoint record into log.

Advantages of using Checkpoints :

- It speeds up data recovery process.
- Most of the dbms products automatically checkpoints themselves.
- Checkpoint records in log file is used to prevent unnecessary redo operations.
- Since dirty pages are flushed out continuously in the background, it has very low overhead and can be done frequently.

Real-Time Applications of Checkpoints :

- Whenever an application is tested in real-time environment that may have modified the database, it is verified and validated using checkpoints.
- Checkpoints are used to create backups and recovery prior to applying any updates in the database.
- The recovery system is used to return the database to the checkpoint state.

Q.4) Explain Conflict serializability, View serializability

ANS:

Conflict Serializability

- Two schedules are said to be conflict equivalent if all the conflicting operations in both the schedule get executed in the same order. If a schedule is a conflict equivalent to its serial schedule then it is called Conflict Serializable Schedule.
- If a schedule is view serializable then it may or may not be conflict serializable.
- Conflict equivalence can be easily achieved by reordering the operations of two transactions therefore, Conflict Serializability is easy to achieve.
- For a transaction T1 writing a value A that no one else reads but later some other transactions say T2 write its own value of A, W(A) cannot be placed under positions where it is never read.

View Serializability

- Two schedules are said to be view equivalent if the order of initial read, final write and update operations is the same in both the schedules. If a schedule is view equivalent to its serial schedule then it is called View Serializable Schedule.
- If a schedule is conflict serializable then it is also view serializable schedule.
- Viewequivalence is rather difficult to achieve as both transactions should perform similar actions in a similar manner. Thus, View Serializability is difficult to achieve.
- If a transaction T1 writes a value A that no other transaction reads (because later some other transactions say T2 writes its own value of A) W(A) can be placed in positions of the schedule where it is never read.