



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS

III YEAR / V SEMESTER

Unit V- CONCURRENCY CONTROL AND RECOVERY SYSTEM

**Topic : TIME-STAMP BASED PROTOCOLS,
VALIDATION BASED PROTOCOLS**



TIMESTAMP BASED PROTOCOLS IN DBMS

Timestamp-based protocols in dbms are used to **order the transaction in ascending order of their creation time.**

The creation time is the **system time or a logical counter.**

The transaction which is created first or you can say **older transactions are given high priority** over new transactions.



SCALER
Topics

For example, if there are two transactions T1 and T2.

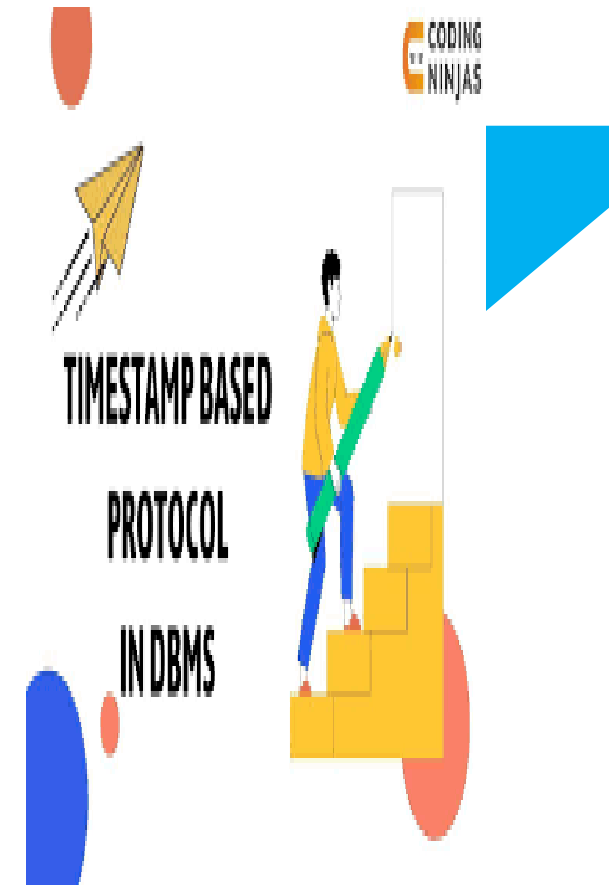
T1 enters the system at 008 and T2 enters the system at 009 then T1 is given priority over T2.



How a Timestamp Ordering Protocol Works?

Timestamp-based protocols in dbms order the transaction according to **their transaction timestamps.**

A schedule that is ordered in the serial order of their transaction timestamp is the only serializable schedule equivalent to the timestamp-ordered-based transaction schedule.





W_TS(X) (write timestamp) is the largest timestamp of any transaction that executed **write(X) successfully.**

R_TS(X) (read timestamp) is the largest timestamp of any transaction that executed **read(X) successfully.**



Advantages

- ✓ No deadlock occurs when timestamp ordering protocol is used as no transaction waits.
- ✓ No older transaction waits for a longer period of time so the protocol is free from deadlock.
- ✓ Timestamp based protocol in dbms ensures that there are no conflicting items in the transaction execution.



Disadvantages

Timestamp-based protocols in dbms may not be **cascade(sequence) free or recoverable.**

In timestamp based protocol in dbms there is a possibility of **starvation of long transactions** if a sequence of conflicting short transactions causes repeated restarting of the long transaction.



Conclusion

Timestamp-based protocols in dbms are used to order the database transaction in order of their creation timestamp.

Timestamp-based protocols in dbms make the transaction serializable.

Transactions based on Timestamp-based protocols in dbms are deadlock-free.

Timestamp-based protocol-based transactions may not be cascade free or recoverable.

The conflicting operation in Timestamp-based protocol-based transactions should not violate the order.

CONCURRENCY CONTROL AND RECOVERY

SYSTEM / 19SB504/DATABASE

MANAGEMENT

SYSTEMS/Mr.R.Kamalakkannan/CSE-



Validation Based Protocol in DBMS



www.educba.com



Validation Based Protocol is also called **Optimistic Concurrency Control Technique**.

This protocol is used in DBMS (Database Management System) for **avoiding concurrency in transactions**.

It is called optimistic because of the assumption it makes, i.e. very less interference occurs, therefore, there is no need for checking while the transaction is executed.



In this technique, **no checking** is done while the transaction is been executed.

Until the transaction end is reached **updates** in the transaction are **not applied directly to the database**.

All updates are applied to local copies of data items kept for the transaction.

At the end of transaction execution, while execution of the transaction, a **validation phase checks whether any of transaction updates violate** serializability.



If there is **no violation of serializability** the transaction is **committed and the database is updated**; or else, the transaction is updated and then **restarted**.

Optimistic Concurrency Control is a **three-phase protocol**. The three phases for validation based protocol:

1. Read Phase
2. Validation Phase
3. Write Phase



1. Read Phase

Values of committed data items from the database can be **read by a transaction**. Updates are only applied to **local data versions**.

2. Validation Phase

Checking is performed to make sure that there is **no violation** of serializability when the transaction updates are applied to the database.

3. Write Phase:

On the **success of the validation phase**, the transaction updates are **applied to the database**, otherwise, the updates are **discarded and the transaction is slowed down**.



Example

Consider a distributed database with two nodes, A and B. A transaction T is initiated that involves updating records in both nodes.

Prepare Phase

Coordinator sends a prepare message to A and B.
If A and B are ready to commit, they respond with a "Yes."



Commit Phase

If both A and B respond with "Yes," the coordinator sends a commit message to A and B.

A and B then perform the actual commit operation. If A succeeds but B encounters an error (e.g., disk full), B sends an abort message to the coordinator.



Outcome

If all participants successfully commit, the transaction is committed globally.

If any participant encounters an error or votes to abort during the process, the coordinator instructs all participants to abort, and the transaction is rolled back.



Thank you