

PRESENTATION

Concurrency and its Problems

Submitted to : Mr.Vishal patiyal

Submitted by : Nikita Kumari

CONTENTS

- Introduction
- Why Concurrency
- Problems of Concurrency
- Types of Concurrency
- Conclusion

INTRODUCTION

- Concurrency means multiple transactions executing simultaneously.
- Improves system performance and resource utilization.
- Common in multi-user environments (e.g., banking, online reservations).

WHY CONCURRENCY

- Better CPU utilization.
- Faster response time for users.
- Enables parallel processing.
- Allows simultaneous access to data.

PROBLEMS OF CONCURRENCY

- Lost Update Problems
- Temporary Update(dirty Read)
- Incorrect Summary problem
- Unrepeatable Read
- Phantom Read



LOST UPDATE PROBLEM

- Two transactions update the same data without awareness of each other.
- One update is overwritten/lost.
- Example: T1 and T2 read the same value and both write different updated values.



TEMPORARY UPDATE

- A transaction reads data updated by another uncommitted transaction.
- If the first transaction rolls back, the second has read invalid data.



UNREAPTABLE READ

- A transaction reads the same row twice and gets different data each time.
- Another transaction has modified the row between reads.



PHANTOM READ

- A transaction reads a set of rows based on a condition.
- Another transaction inserts/deletes rows affecting the result.
- On re-execution, the result set changes unexpectedly.

TYPES OF CONCURRENCY CONTROL

- Lock Based Protocol
- Timestamp-Protocols
- optimistic Concurrency Control
- Multiversion Concurrency Control(MVCC)



LOCK-BASED PROTOCOLS

- Transactions acquire locks before accessing data.
- Types: Shared Lock (Read) and Exclusive Lock (Write)
- Follow Two-Phase Locking (2PL) for serializability;



TIMESTAMP-BASED PROTOCOLS

- Every transaction gets a timestamp.
- Ensures conflict-free execution based on timestamps.
- Follows rules for read/write operations to maintain order



OPTIMISTIC CONCURRENCY CONTROL

- No locks; assumes conflicts are rare.
- Works in 3 phases: Read, Validate, Write
- Suitable for low-conflict environments.



MULTIVERSION CONCURRENCY CONTROL(MVCC)

- Keeps multiple versions of data items.
- Readers don't block writers and vice versa.
- Used in databases like PostgreSQL and Oracle.

CONCLUSION

- Concurrency is essential for performance and efficiency.
- Must handle problems to ensure data consistency.
- Use appropriate control techniques for safe concurrent access.