LET'S START WITH DBMS :)

Concurrency control mechanisms

Concurrency control -> It is a process of mananging multiple users access and modification in data simultaneously in shared or multi-user database systems.

How it helps?
1. **Data Consistency**: Ensures that data remains accurate and reliable despite concurrent access.
2. **Isolation**: Maintains the isolation property of transactions, so the outcome of a transaction is not affected by other concurrently executing transactions.
3. **Serializability**: Ensures that the result of concurrent transactions is the same as if the transactions had been executed serially

LET'S START WITH DBMS :)

Concurrency control mechanisms

Concurrency control mechanisms are techniques used in DBMS to ensure that transactions are executed concurrently without leading to inconsistencies in the database. In serializability we check if a schedule is serializable or not but in concurrency control we use certain techniques to make a schedule serializable.

Why its needed?

- **Lost Updates:** When two or more transactions update the same data simultaneously, one of the updates might be lost. For example, if two users modify the same record at the same time, the changes made by one user could overwrite the changes made by the other. (WW)

- **Dirty Reads:** When a transaction reads data that has been modified by another transaction but not yet committed. If the first transaction rolls back, the other transaction will have read invalid data. (WR)

LET'S START WITH DBMS :)

Concurrency control mechanisms

- **Uncommitted Dependency (Dirty Writes):** When a transaction modifies data that has been read by another transaction, leading to inconsistencies if one of the transactions rolls back.

- **Inconsistent Retrievals (Non-repeatable Reads):** Happens when a transaction reads the same data multiple times and gets different values each time because another transaction is modifying the data in between the reads.

- **Phantom Reads:** Occurs when a transaction reads a set of rows that satisfy a condition, but another transaction inserts or deletes rows that affect the set before the first transaction completes. This results in the first transaction reading different sets of rows if it re-executes the query.

LET'S START WITH DBMS :)

Concurrency control mechanisms

Some common concurrency control mechanisms are :

- **Lock-Based Protocols**(2PL, Strict 2PL, Rigorous 2PL , Conservative 2PL)
- **Timestamp-Based Protocols** (Basic Timestamp Ordering (TO))
- **Optimistic Concurrency Control** (OCC)
- **Multiversion Concurrency Control** (MVCC)