Transaction Isolation Levels in DBMS

Prerequisite – Concurrency control in DBMS, ACID Properties in DBMS

As we know, in order to maintain consistency in a database, it follows ACID properties. Among these four properties (Atomicity, Consistency, Isolation, and Durability) Isolation determines how transaction integrity is visible to other users and systems. It means that a transaction should take place in a system in such a way that it is the only transaction that is accessing the resources in a database system.

Isolation levels define the degree to which a transaction must be isolated from the data modifications made by any other transaction in the database system. A transaction isolation level is defined by the following phenomena:

- **Dirty Read** – A Dirty read is a situation when a transaction reads data that has not yet been committed. For example, Let's say transaction 1 updates a row and leaves it uncommitted, meanwhile, Transaction 2 reads the updated row. If transaction 1 rolls back the change, transaction 2 will have read data that is considered never to have existed.
- **Non Repeatable read** – Non Repeatable read occurs when a transaction reads the same row twice and gets a different value each time. For example, suppose transaction T1 reads data. Due to concurrency, another transaction T2 updates the same data and commit, Now if transaction T1 rereads the same data, it will retrieve a different value.
- **Phantom Read** – Phantom Read occurs when two same queries are executed, but the rows retrieved by the two, are different. For example, suppose transaction T1 retrieves a set of rows that satisfy some search criteria. Now, Transaction T2 generates some new rows that match t search criteria for transaction T1. If transaction T1 re-executes t

Based on these phenomena, The SQL standard defines four isolation levels:

1. **Read Uncommitted** – Read Uncommitted is the lowest isolation level. In this level, one transaction may read not yet committed changes made by other transactions, thereby allowing dirty reads. At this level, transactions are not isolated from each other.
2. **Read Committed** – This isolation level guarantees that any data read is committed at the moment it is read. Thus it does not allow dirty read. The transaction holds a read or write lock on the current row, and thus prevents other transactions from reading, updating, or deleting it.
3. **Repeatable Read** – This is the most restrictive isolation level. The transaction holds read locks on all rows it references and writes locks on referenced rows for update and delete actions. Since other transactions cannot read, update or delete these rows, consequently it avoids non-repeatable read.
4. **Serializable** – This is the highest isolation level. A *serializable* execution is guaranteed to be serializable. Serializable execution is defined to be an execution of operations in which concurrently executing transactions appears to be serially executing.

The Table given below clearly depicts the relationship between isolation levels, read phenomena, and locks:

Isolation Level	Dirty reads	Non-repeatable reads	Phantoms
Read Uncommitted	May occur	May occur	May occur
Read Committed	Don't occur	May occur	May occur
Repeatable Read	Don't occur	Don't occur	May occur
Serializable	Don't occur	Don't occur	Don't occur

Anomaly Serializable is not the same as Serializable. That is, it is necessary,

phenomena types.

Transaction isolation levels are used in database management systems (DBMS) to control the level of interaction between concurrent transactions.

The four standard isolation levels are:

Read Uncommitted: This is the lowest level of isolation where a transaction can see uncommitted changes made by other transactions. This can result in dirty reads, non-repeatable reads, and phantom reads.

Read Committed: In this isolation level, a transaction can only see changes made by other committed transactions. This eliminates dirty reads but can still result in non-repeatable reads and phantom reads.

Repeatable Read: This isolation level guarantees that a transaction will see the same data throughout its duration, even if other transactions commit changes to the data. However, phantom reads are still possible.

Serializable: This is the highest isolation level where a transaction is executed as if it were the only transaction in the system. All transactions must be executed sequentially, which ensures that there are no dirty reads, non-repeatable reads, or phantom reads.

The choice of isolation level depends on the specific requirements of the application. Higher isolation levels offer stronger data consistency but can also result in longer lock times and increased contention, leading to decreased concurrency and performance. Lower isolation levels provide more concurrency but can result in data inconsistencies.

In addition to the standard isolation levels, some DBMS may also support additional custom isolation levels or features such as snapshot isolation and multi-version concurrency control (MVCC) that provide alternative solutions to the problems addressed by the standard isolation levels.

Advantages of Transaction Isolation Levels:

Improved concurrency: Transaction isolation levels can improve concurrency

each other.

Control over data consistency: Isolation levels provide control over the level of data consistency required by a particular application.

Reduced data anomalies: The use of isolation levels can reduce data anomalies such as dirty reads, non-repeatable reads, and phantom reads.

Flexibility: The use of different isolation levels provides flexibility in designing applications that require different levels of data consistency.

Disadvantages of Transaction Isolation Levels:

Increased overhead: The use of isolation levels can increase overhead because the database management system must perform additional checks and acquire more locks.

Decreased concurrency: Some isolation levels, such as Serializable, can decrease concurrency by requiring transactions to acquire more locks, which can lead to blocking.

Limited support: Not all database management systems support all isolation levels, which can limit the portability of applications across different systems.

Complexity: The use of different isolation levels can add complexity to the design of database applications, making them more difficult to implement and maintain.

Last Updated : 24 Apr, 2023