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 IV- **TRANSACTIONS MANAGEMENT**

Topic :**ISOLATION LEVELS**

ISOLATION LEVELS

- ✓ Isolation levels in a Database Management System (DBMS) define the level of isolation or **separation between concurrent transactions**, specifying how changes made by one transaction are visible to other concurrent transactions.
- ✓ There are **different isolation levels**, each offering a different level of **data consistency and isolation**.
- ✓ The SQL standard defines **four isolation** levels:

1. Read Uncommitted (the lowest isolation level)

- ✓ In this isolation level, transactions are **not isolated from each other at all**.
- ✓ One transaction **can see uncommitted** changes made by another transaction.
- ✓ It allows for the **highest level of concurrency** but the **lowest data consistency**.
- ✓ Generally not recommended for most applications due to the potential for dirty reads, non-repeatable reads, and phantom reads.

2.Read Committed

- ✓ In this isolation level, a **transaction can only see committed changes** made by other transactions.
- ✓ It prevents dirty reads (reading uncommitted data), but it may still allow non-repeatable reads and phantom reads.
- ✓ Read Committed is a good balance between data consistency and concurrency and is suitable for many applications.

3. Repeatable Read:

In this isolation level, a transaction **sees a consistent snapshot of the database** as of the start of the transaction.

It prevents **dirty reads and non-repeatable reads** but may still allow phantom reads.

Provides a **higher level of data consistency** but can **reduce concurrency**.

4. Serializable (the highest isolation level)

- ✓ In this isolation level, **transactions are completely isolated from each other,** as if they were executed one after the other.
- ✓ It prevents **dirty reads, non-repeatable reads, and phantom reads**, offering the highest level of data consistency.
- ✓ However, it can significantly reduce concurrency and may lead to performance issues.

Example

Imagine two transactions, T1 and T2, trying to access a shared bank account for read and write operations. The initial account balance is $1,000.

Read Uncommitted

T1 (Read Uncommitted) reads the account balance (result: $1,000).
T2 (Read Uncommitted) updates the account balance to $900.
T1 (Read Uncommitted) reads the updated balance (result: $900).
In this isolation level, T1 can see the uncommitted changes made by T2.

Read Committed

T1 (Read Committed) reads the account balance (result: $1,000).

T2 (Read Committed) updates the account balance to $900.

T1 (Read Committed) reads the account balance again (result: $1,000).

Read Committed prevents T1 from seeing the uncommitted changes made by T2.

Repeatable Read:

T1 (Repeatable Read) reads the account balance (result: $1,000).

T2 (Repeatable Read) updates the account balance to $900.

T1 (Repeatable Read) reads the account balance again (result: $1,000).

Repeatable Read ensures that T1 sees a consistent snapshot of the database as of the start of the transaction. It doesn't allow T1 to see the change made by T2 during its transaction.

Serializable:

T1 (Serializable) reads the account balance (result: $1,000).

T2 (Serializable) updates the account balance to $900.

T1 (Serializable) reads the account balance again (result: $1,000).

Serializable provides the highest level of isolation, ensuring that T1 is completely isolated from the changes made by T2. T1 always sees the initial state of the account.

Thank You…….