**Problem of Concurrent transaction**

There are three problems for concurrent transaction.

* Dirty read
* Phantom read
* Unrepeatable read

Here is the solution table for theses reads. These isolation levels can solve these problems



Refer this link to understand

<http://javadata.blogspot.in/2011/05/dirty-readphantom-read-and-non.html>

Dirty read :

Two transactions are reading concurrently. T1 reads the result set, count = 10, then T2 modifies or deleted something, for that result set count changed. But T1 is still getting the old value, as T2 is not commited. Read\_commited isolation level will solve this issue

Phantom read :

T1 and T2 two transactions are running concurrently. T1 read the data and go to sleep. T2 update something and go to sleep. T1 woke up and get the previously read data and work with it, whether T2 has commmited its data or not. Only serializable can solve this problem. Where transactions are executed serially rather than parellaly for the same data object.

Unrepeatable read :

If a transaction T1 read a data two times, but as other transactions are running concurrently on the same data updated in middle , Then T1 read two times two different data, We can solve this by repeatable read isolation level. So if a transaction only reads any data, it will read the data just one time and get updated data.