

ISOLATION Levels : Reads

SET TRANSACTION ISOLATION LEVEL <level>

START TRANSACTION

⋮

COMMIT or ROLLBACK

<level>

Serializable

Repeatable Read

Read Committed

Read Uncommitted

Concurrency
weakest
↓
strongest

Consistency
↑

Overhead
↑

Read Uncommitted \equiv Dirty Reads

* non serializable outputs

* non-atomic outputs.

Example Bank reversed

User 1 T1 : Updating Accounts Table

User T2 Reading Accounts

$x = 100$ $y = 50$ initially

T1

T2

Q1: $x = x - 10$

q1: print x

Q2: $y = y + 10$

q2: print y

Q3: commit / rollback

q3: commit

Serializable: T1; T2 x y
 T2; T1 90 60
 100 50

T2: set transaction isolation level read uncommitted;

start transaction

q1

q2

q3

T1: start transaction / commit

<u>Execution order by time</u>	
Q1 Q2 q1 q2	T1; T2
q1 q2 Q1 Q2	T2; T1
Q1 q1 Q2 q2	nonserializable
Q1 q1 q2 Q2	nonserializable
q1 Q1 q2 Q2	nonserializable
q1 Q1 Q2 q2	non-serializable

<u>Output of T2</u>	
90 60	T1; T2
100 50	T2; T1
90 60	T1; T2
90 50	non-serializable non-atomic
100 50	T2; T1
100 60	nonserializable

T1: start transaction / rollback

Only legal result: 100 ; 50

Example

a
1
2

T1: Q1: $a = a + 1$

Q2: $a = 2 * a$

commit

a
2
3

a
4
6

T2: Read uncommitted

q: select a

Output depends on order of execution:

1) q; Q1; Q2 : T2; T1

2) Q1; Q2; q : T1; T2

3) Q1; q; Q2 : non serializable execution order

Show all possible outputs:

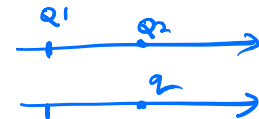
a
2
3

a
2
2

a
1
3

a
4
3

a
2
6



only MySQL
output

won't see this in MySQL, but SQL standard

Read Committed

* non-serializable output

* atomic output

Bank example

u2: T2: set transaction isolation level read committed
start transaction

T1: start transaction / commit

<u>Execution order by time</u>		<u>Output of T2</u>	
Q1 Q2 Q1 Q2	T1; T2	90 60	T1; T2
Q1 Q2 Q1 Q2	T2; T1	100 50	T2; T1
Q1 Q1 Q2 Q2	nonserializable	100 60	nonserializable
Q1 Q1 Q2 Q2	nonserializable	100 50	T2; T1
Q1 Q1 Q2 Q2	nonserializable	100 50	T2; T1
Q1 Q1 Q2 Q2	non-serializable	100 60	non-serializable

T1: with rollback, all outputs look 100; 50

Example:

Salary
10
20

sum: 30

T1: Q1: update Emp set salary = salary + 100;
Q2: commit

sum = 230

Salary
110
120

T2: Read committed

q_1 : select sum(salary) from Emp;
 q_2 : select sum(salary) from Emp;

<u>Execution orders</u>						<u>Output</u>		
$T_1; T_2$:	Q_1	Q_2	q_1	q_2	230	230	$T_1; T_2$
$T_2; T_1$:	q_1	q_2	Q_1	Q_2	30	30	$T_2; T_1$
non serial executions	:	Q_1	q_1	Q_2	q_2	30	230	
		q_1	Q_1	Q_2	q_2	30	230	
		q_1	Q_1	q_2	Q_2	30	30	$T_2; T_1$
		Q_1	q_1	q_2	Q_2	30	30	$T_2; T_1$

if T_1 had rollback; then all outputs 30;30