TP Book Study

Chapter 7
Isolation Level
: Locking Basic and Practice

Jong-Hyeok Park
akindo19@gmail.com

Introduction

Life of Transaction

Start TX

READ

WRITE

Commit

Abort

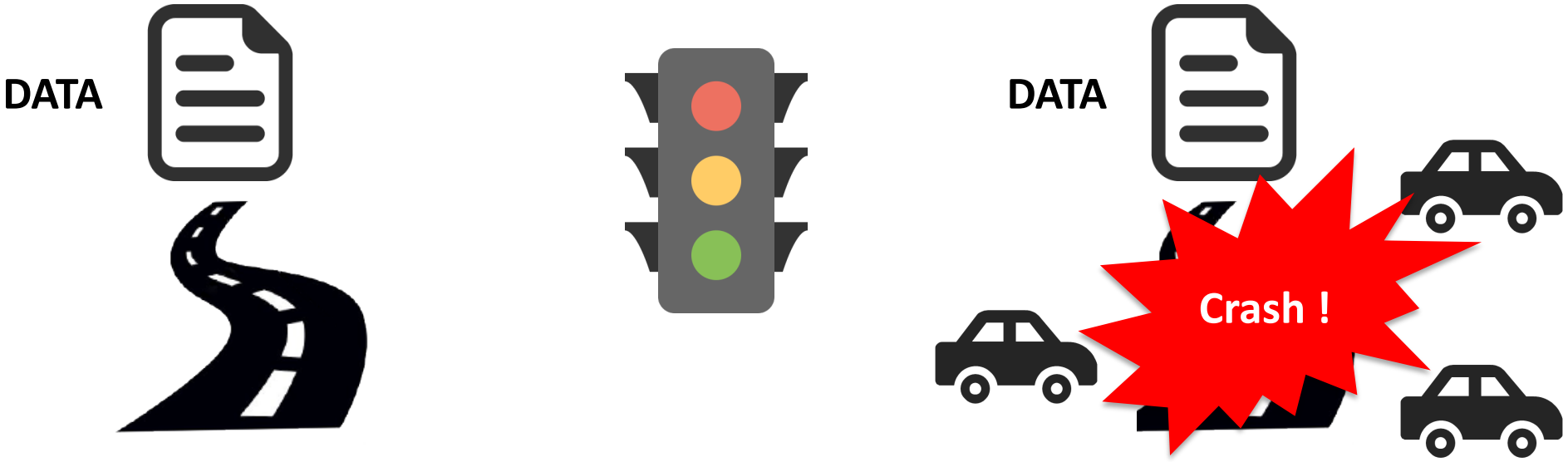
Rollback

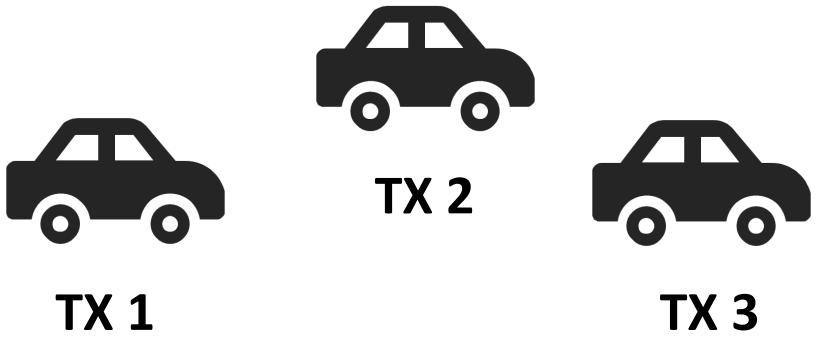
End TX

Introduction

Why Locking ?

Concurrency Control !



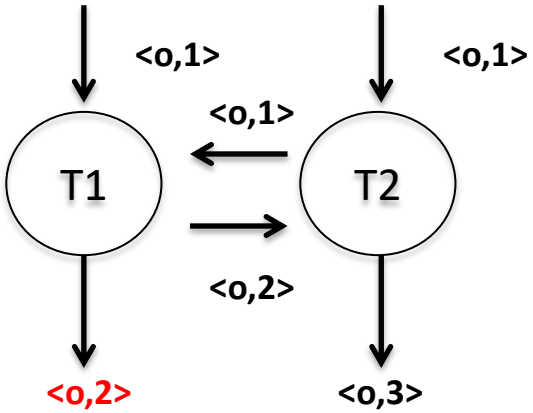


Locking Basic

Bad Dependency

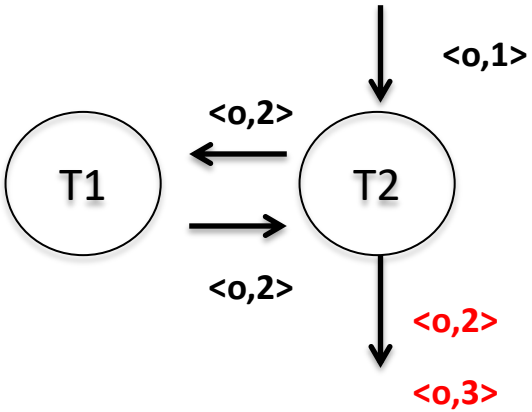
1. Lost Update

T2 READ <o , 1>
T1 WRITE <o , 2>
T2 WRITE <o , 3>



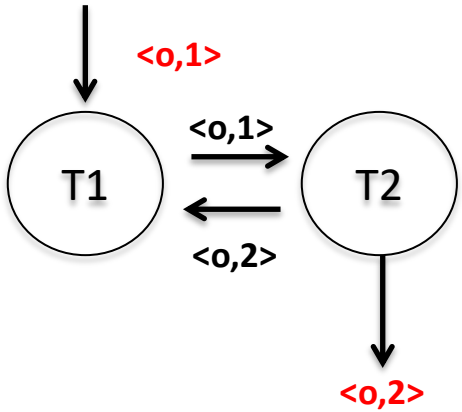
2. Dirty Read

T2 WRITE <o , 2>
T1 READ <o , 2>
T2 WRITE <o , 3>



3. Unrepeatable Read

T1 READ <o , 1>
T2 WRITE <o , 2>
T1 READ <o , 2>



+ Phantom Problem …

Dirty Read

```
[Mon Dec  7 15:10:59 2015][chunge] > show variables like 'tx_isolation';
+---------------+------------------+
| Variable_name | Value            |
+---------------+------------------+
| tx_isolation  | READ-UNCOMMITTED |
+---------------+------------------+
1 row in set (0.00 sec)
 [Mon Dec  7 15:11:05 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)
[Mon Dec  7 15:11:09 2015][chunge] > insert into chunge.chung values(4);
Query OK, 1 row affected (0.00 sec)
[Mon Dec  7 15:11:18 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.01 sec)
[Mon Dec  7 15:11:22 2015][chunge] > rollback;
Query OK, 0 rows affected (0.00 sec)
[Mon Dec  7 15:11:32 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)
```

ROLLBACK !

```
[Mon Dec  7 15:10:59 2015][chunge] > show variables like 'tx_isolation';
+---------------+------------------+
| Variable_name | Value            |
+---------------+------------------+
| tx_isolation  | READ-UNCOMMITTED |
+---------------+------------------+
1 row in set (0.00 sec)
[Mon Dec  7 15:11:05 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)


[Mon Dec  7 15:11:18 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.01 sec)


[Mon Dec  7 15:11:32 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)
```

Non-Repeatable Read

```
[Mon Dec  7 15:25:12 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    5 |
+------+
2 rows in set (0.00 sec)
```

UPDATE !

```
[Mon Dec  7 15:25:26 2015][chunge] > update chunge.chung set no=4 where no=5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

[Mon Dec  7 15:25:51 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.00 sec)
```

```
[Mon Dec  7 15:25:12 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    5 |
+------+
2 rows in set (0.00 sec)

[Mon Dec  7 15:25:19 2015][chunge] >
[Mon Dec  7 15:25:51 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.00 sec)
```

Phantom Read

```
[Mon Dec  7 15:19:23 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)
[Mon Dec  7 15:19:26 2015][chunge] > insert into chunge.chung values(5);
Query OK, 1 row affected (0.00 sec)

[Mon Dec  7 15:19:32 2015][chunge] > commit;
Query OK, 0 rows affected (0.00 sec)

[Mon Dec  7 15:19:35 2015][chunge] >
[Mon Dec  7 15:19:39 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    5 |
+------+
2 rows in set (0.00 sec)
```

INSERT !

```
[Mon Dec  7 15:19:16 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
+------+
1 row in set (0.00 sec)

[Mon Dec  7 15:19:26 2015][chunge] >
[Mon Dec  7 15:19:26 2015][chunge] >
[Mon Dec  7 15:19:39 2015][chunge] > select * from chunge.chung;
+------+
| no   |
+------+
|    3 |
|    5 |
+------+
2 rows in set (0.00 sec)
```

Why ? Lock only the record not the “gap”

Introduction

Why Locking ?

- Create Lock
- First Come First Serve
- Owner or Waiter

TX 1

TX 2

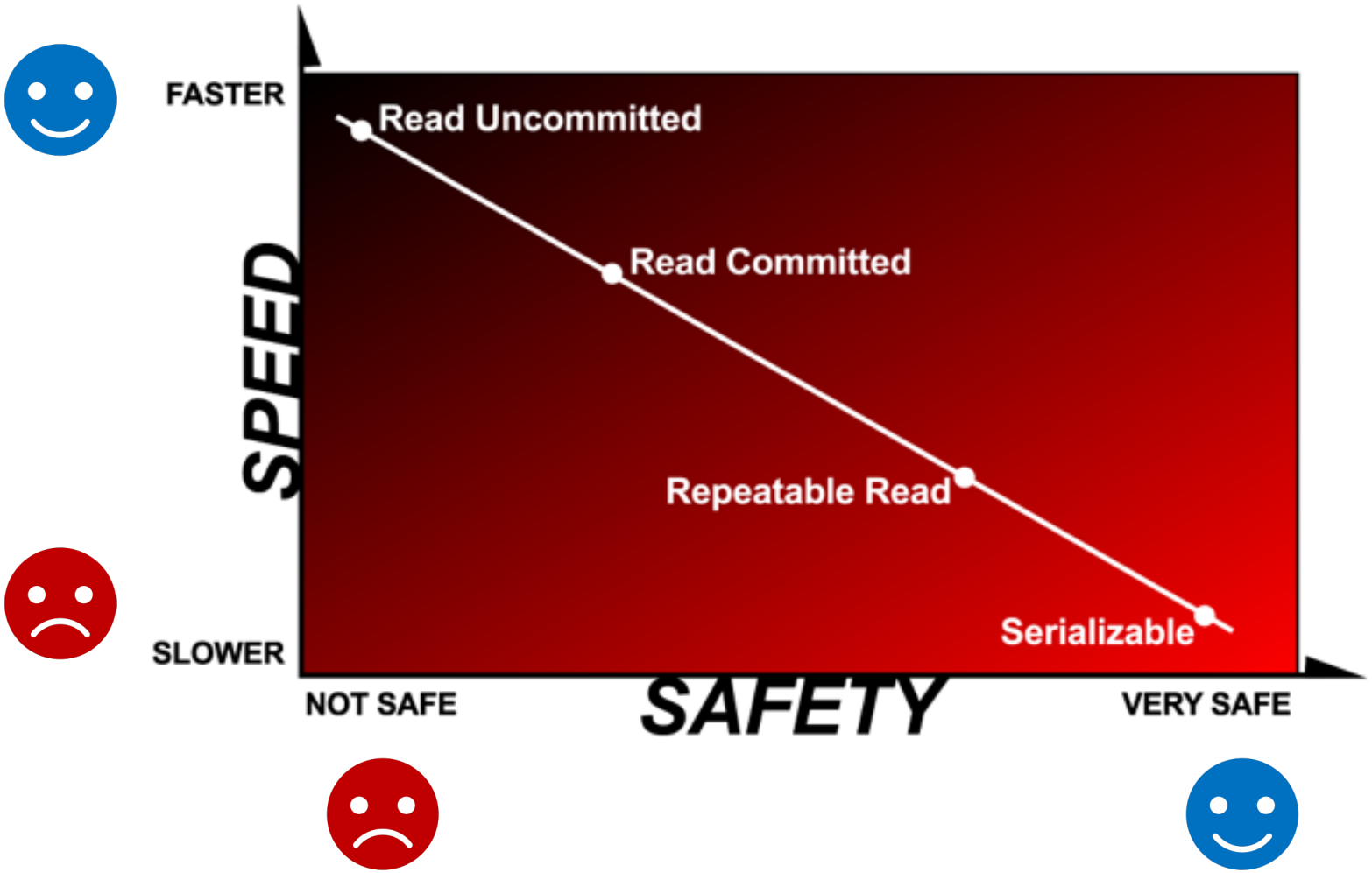
TX 3

TX 1

TX 2

TX 3

Introduction



http://archive.oreilly.com/oreillyschool/courses/dba1/dba105.html

Locking Basic

Transaction Isolation Level

+ SERIALIZABLE

1. Lost Update

T2	READ	<o , 1>
T1	WRITE	<o , 2>
T2	WRITE	<o , 3>

READ UNCOMMITTED

- **Dirty Read**
- **Even if Single Statement**

2. Dirty Read

T2	WRITE	<o , 2>
T1	READ	<o , 2>
T2	WRITE	<o , 3>

READ COMMITTED

- **Non-Repeatable Read**
- **ReadView Create**
  @ Every statement start

3. Unrepeatable Read

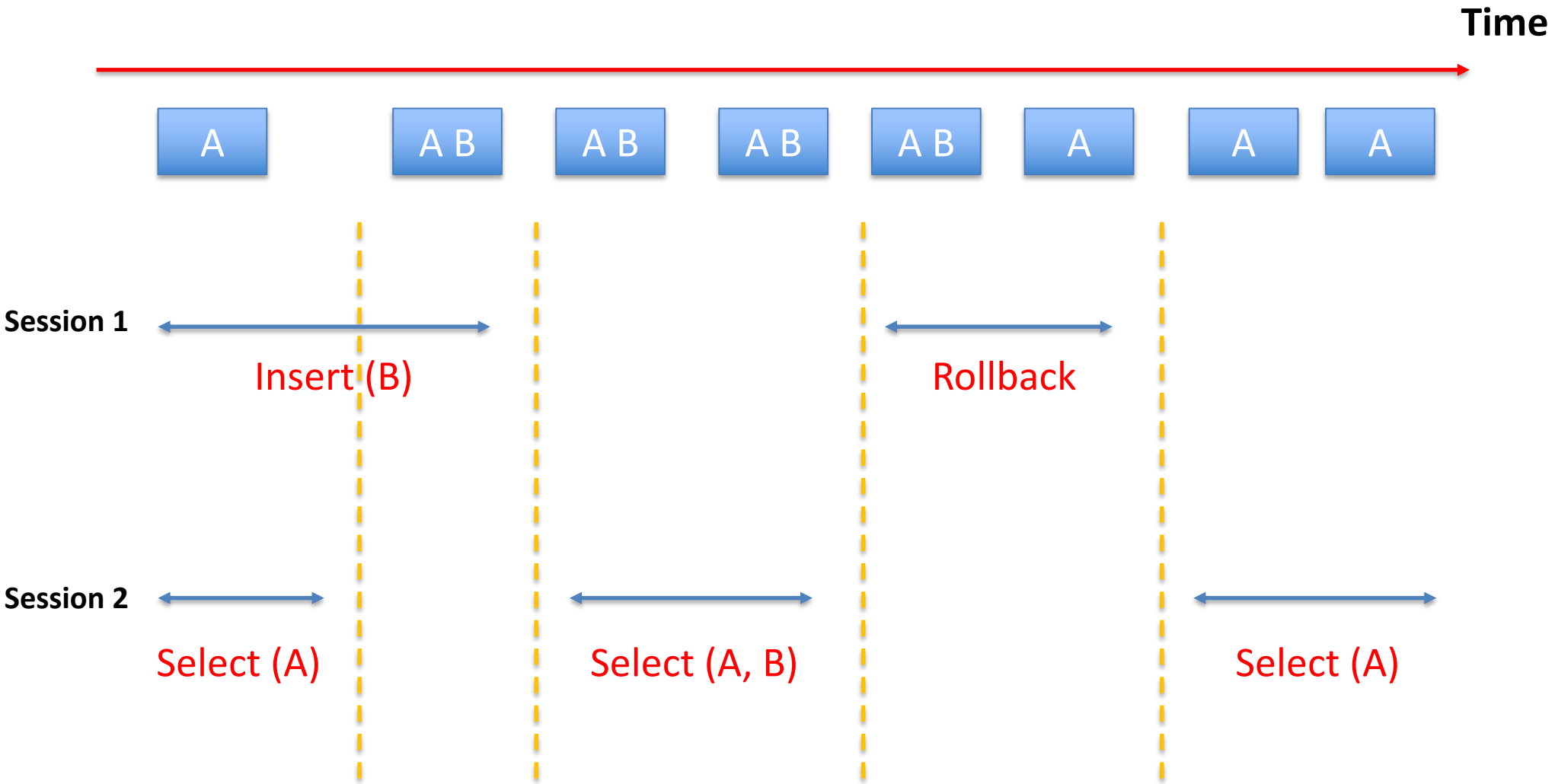
T1	READ	<o , 1>
T2	WRITE	<o , 2>
T1	READ	<o , 2>

REPEATABLE READ

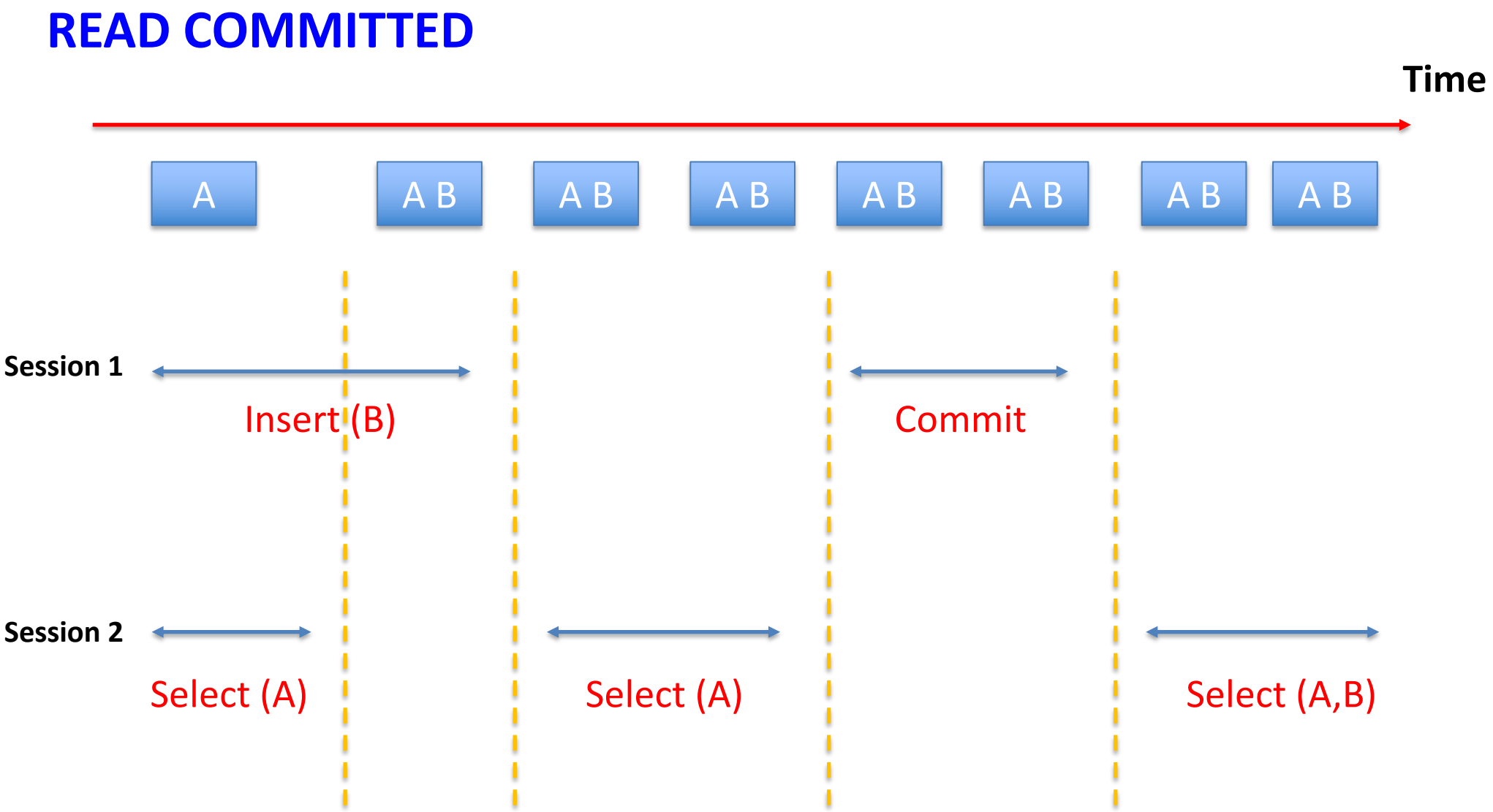
- **Default**
- **ReadView Create**
  @ TX start (Actually, First Read)
- **Locking Read**

Isolation Level in MySQL

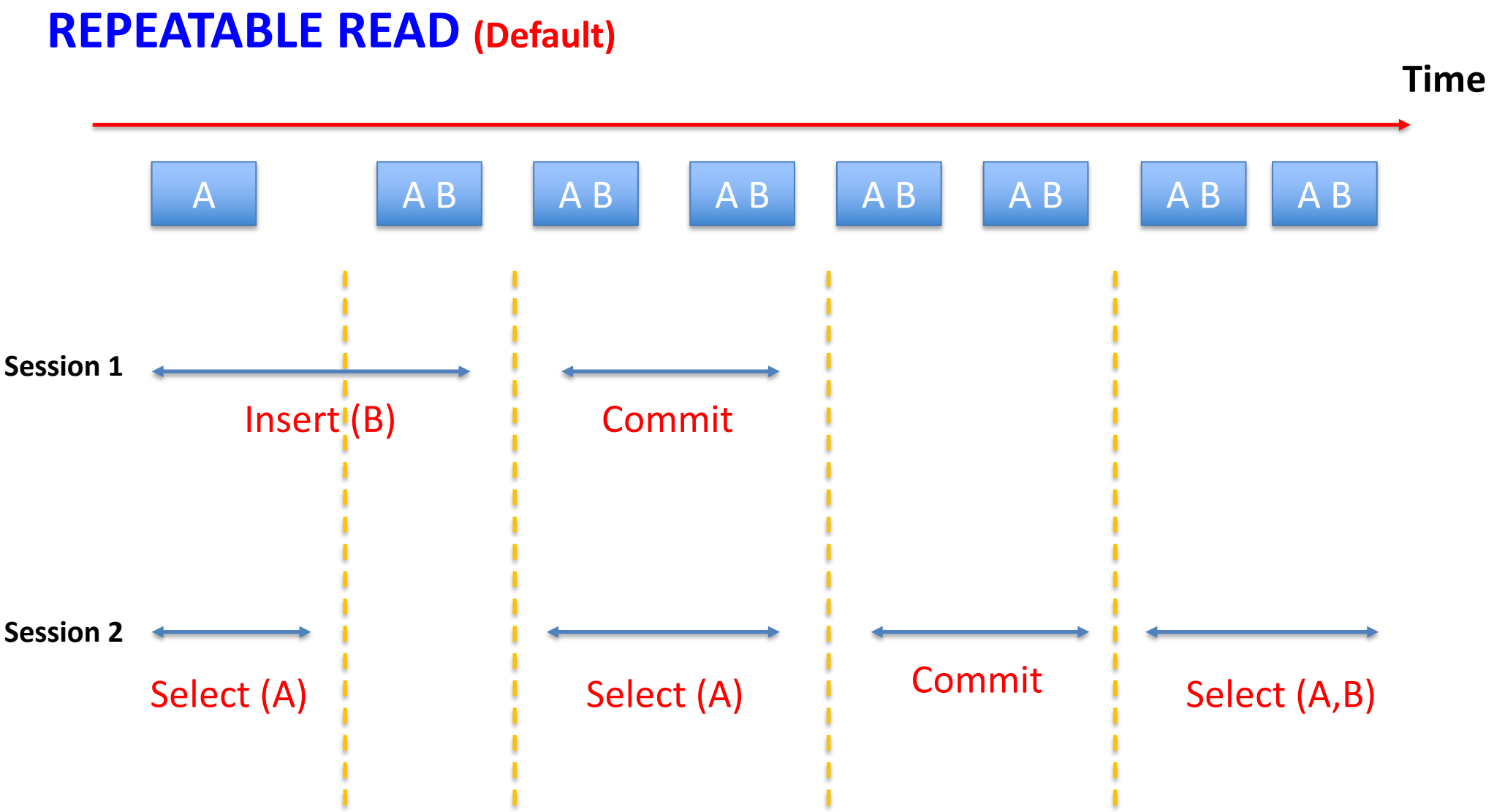
READ UNCOMMITTED



Isolation Level in MySQL

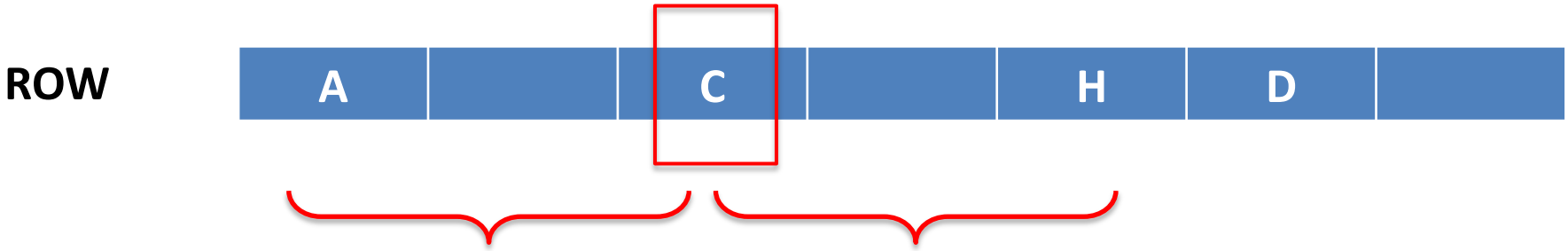


Isolation Level in MySQL



Isolation Level in MySQL

REPEATABLE READ (Default)



SERIALIZABLE

SELECT … FROM TBL

SELECT … FROM TBL
LOCK IN SHAER MODE

Phantom Write

In **REPEATABLE READ** in MySQL

Session #1	Session #2
SELECT * FROM TBL **[empty]**	
	INSERT TBL (1, “AA”) INSERT TBL (2, “BB”) COMMIT
SELECT * FROM TBL **[empty]**	
UPDATE SET “XX” WHERE ID = 1 **[UPDATE SUCCESS!] ?!** SELECT * FROM TBL **[1, “XX”]** COMMIT	
SELECT * FROM TBL **[1, “XX”]** **[2, “BB”]**	

Phantom Read OK !

Phantom Write NO !

Summary

Isolation Level	Dirty Read	Repeatable Read	Phantom
READ UNCOMMITTED	O	O	O
READ COMMITTED	X	O	O
REPEATABLE READ	X	X	READ : X Write : O
SERIALIZABLE	X	X	X