

# T-SQL-III

## Snapshot Isolation

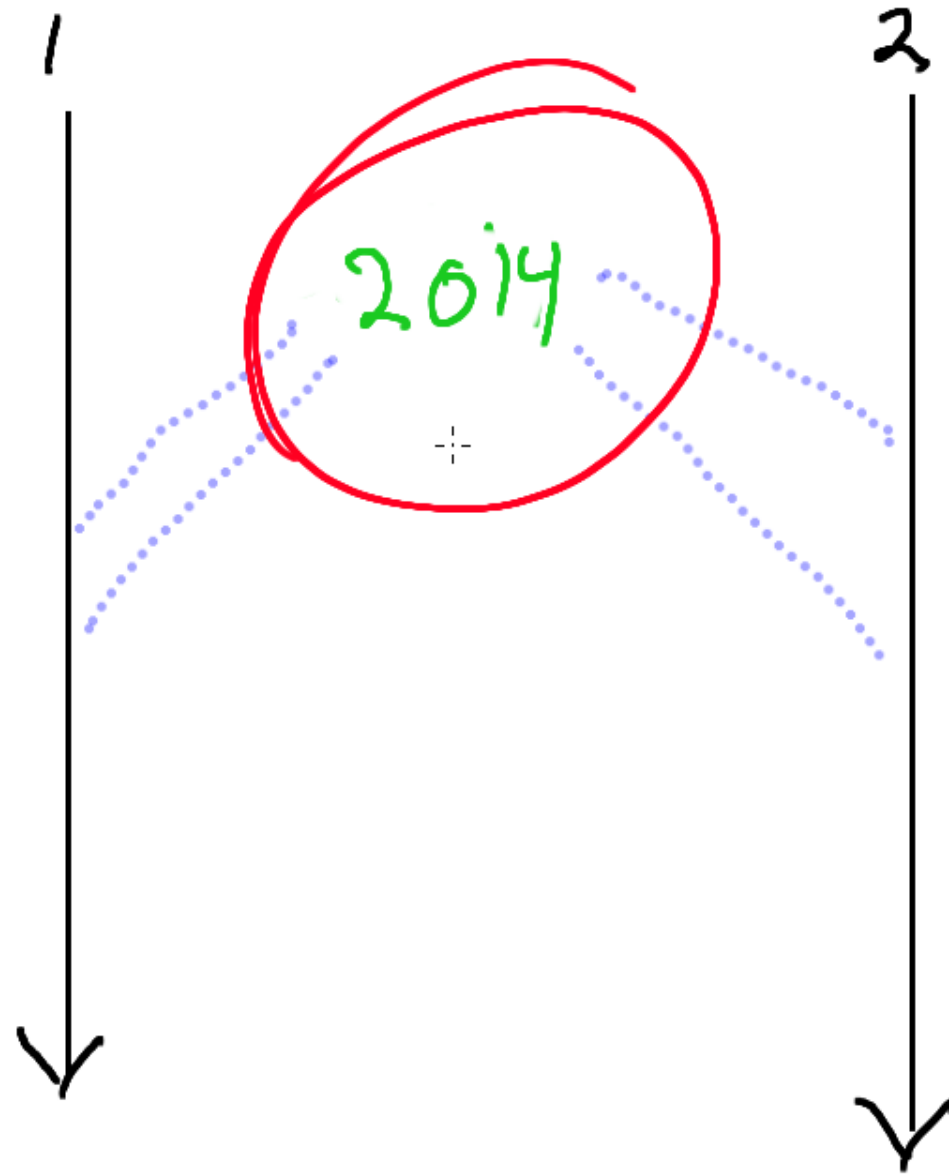


# Snapshot Isolation

- **Transaction**
  - all or nothing
- **Isolation**
  - complete -> almost none

Proct  
R-M-W

$$\begin{array}{r} 2013 \\ +1 \\ \hline 2014 \end{array}$$



$$\begin{array}{r} 2013 \\ +1 \\ \hline 2014 \end{array}$$

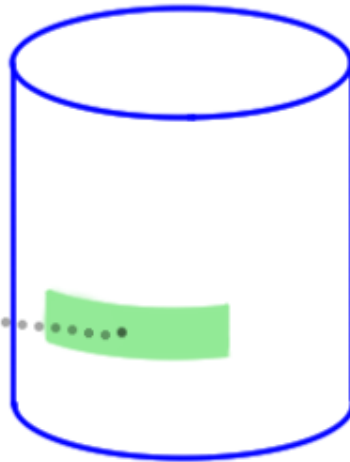
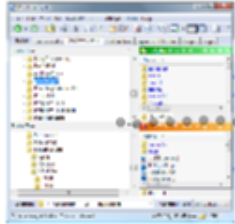
Proct  
R.M.H

2:04  
85.1

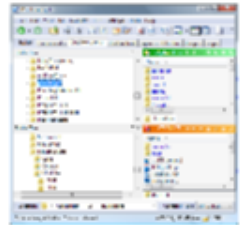
Time 2:04  
Temp 93.5



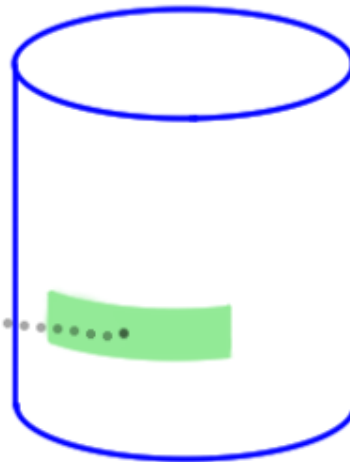
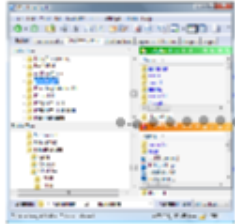
1



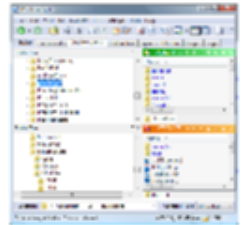
2



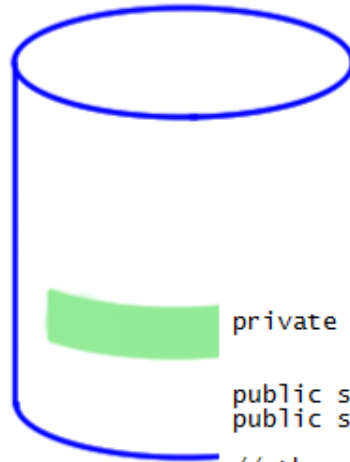
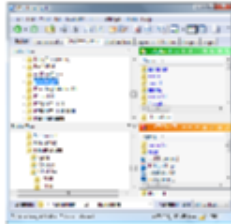
1



2



1



private static object door;

public static string Time;  
public static string Temp;

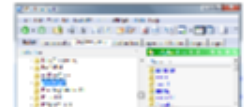
// thread 1  
public void TakeTemp(string time, string temp)

```
{
    lock(door)
    {
        Time = time;
        Temp = temp;
    }
}
```

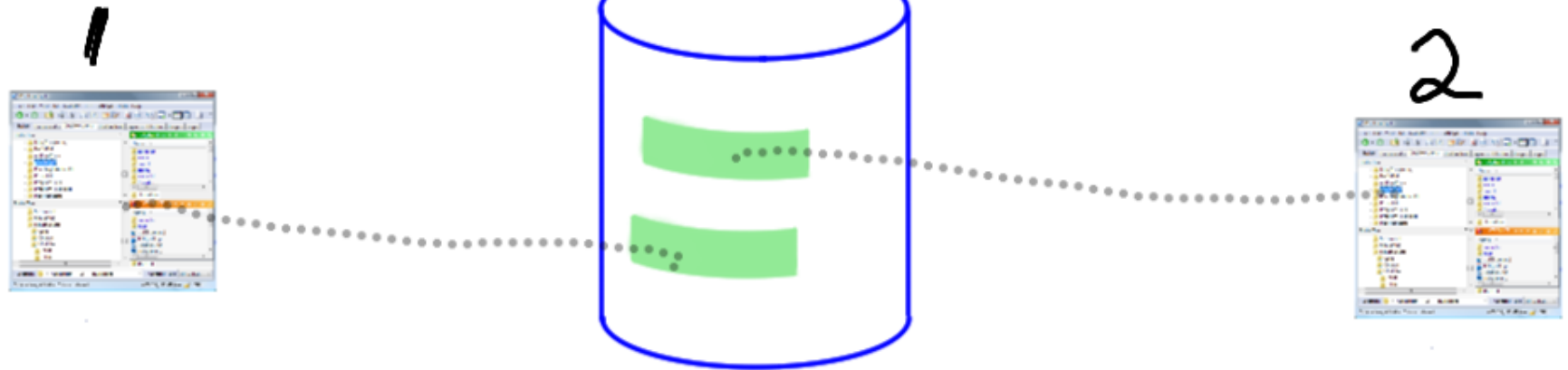
// thread 2  
public void DisplayTemp()

```
{
    lock (door)
    {
        Console.WriteLine("Time: {0} Temp:{1}", Time, Temp);
    }
}
```

2



# Serializable

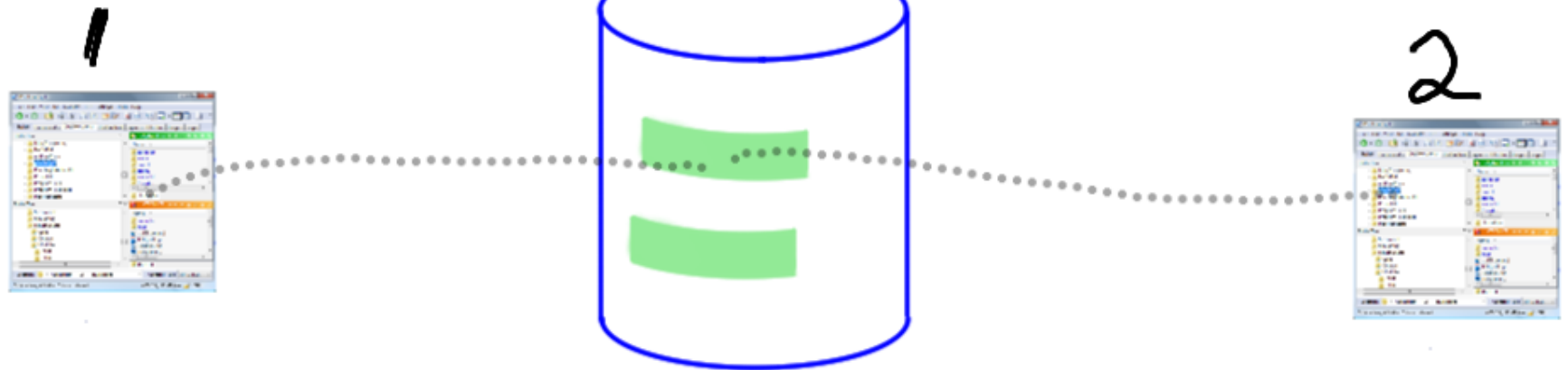


```
begin tran  
update data set temp=84.1;  
update date set time=1:04;  
commit tran
```

2



# Serializable



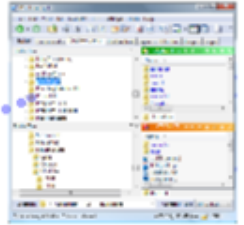
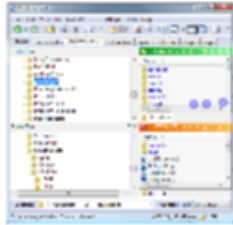
```
begin tran  
update data set temp=84.1;  
update date set time=1:04;  
commit tran
```

2

# Serializable

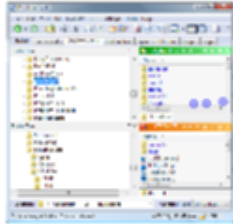
1

2

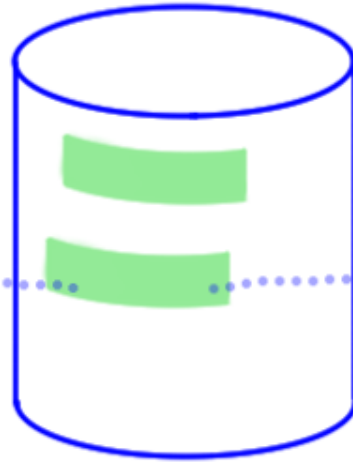
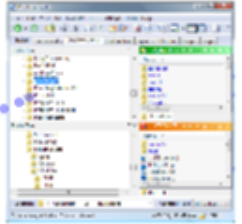


# Serializable

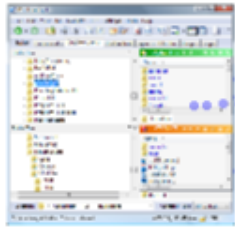
1



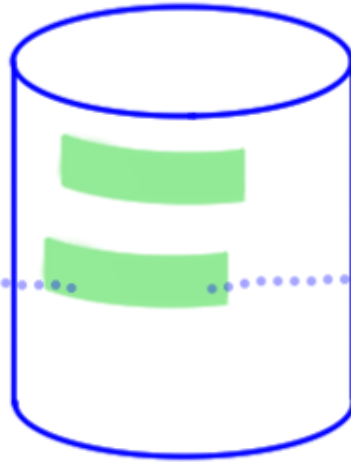
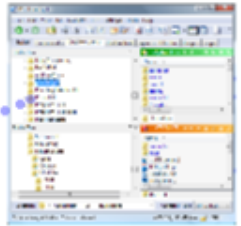
2



1

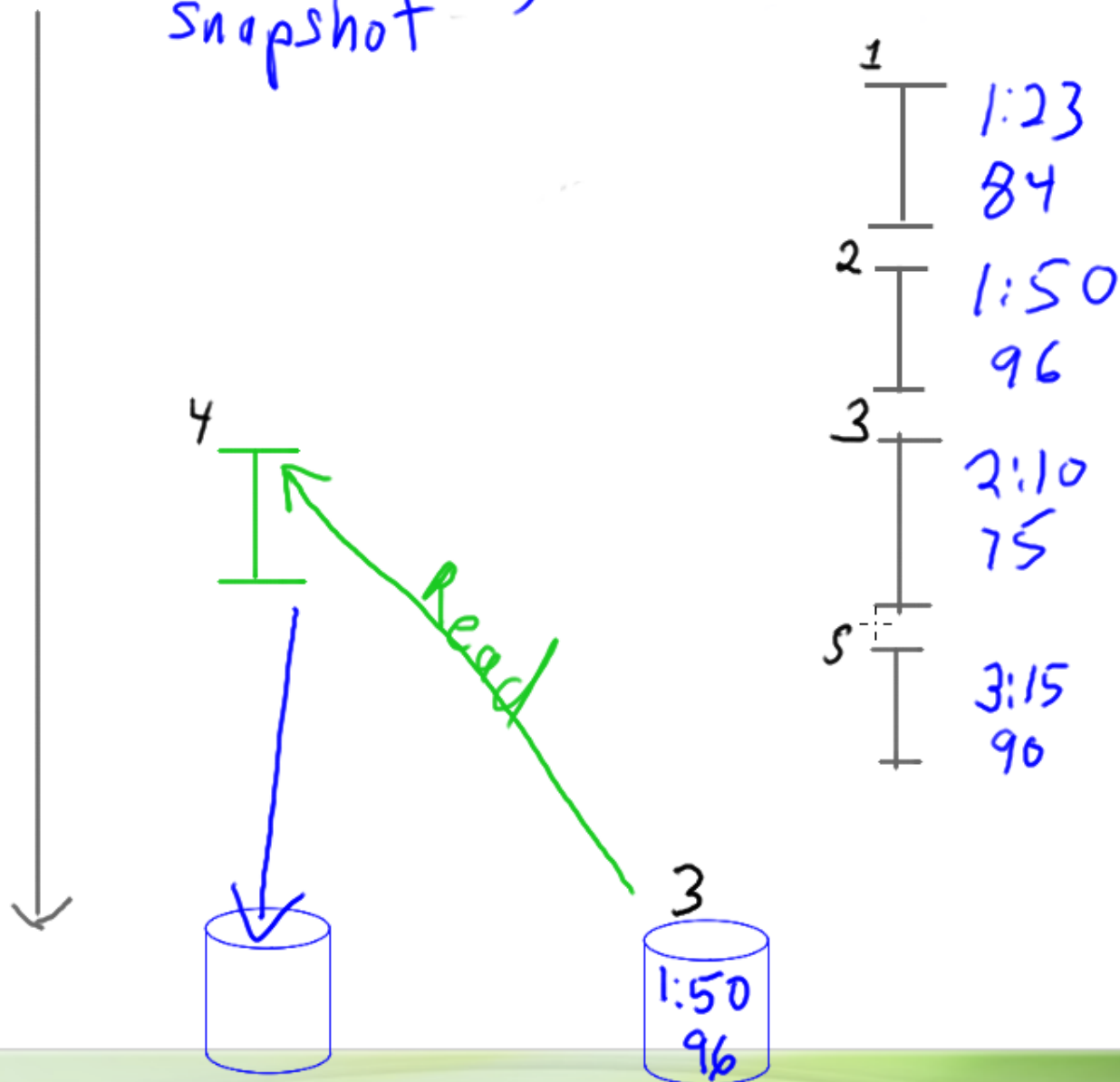


2



Conn1 (Read)  
Snapshot

Conn2 (write)

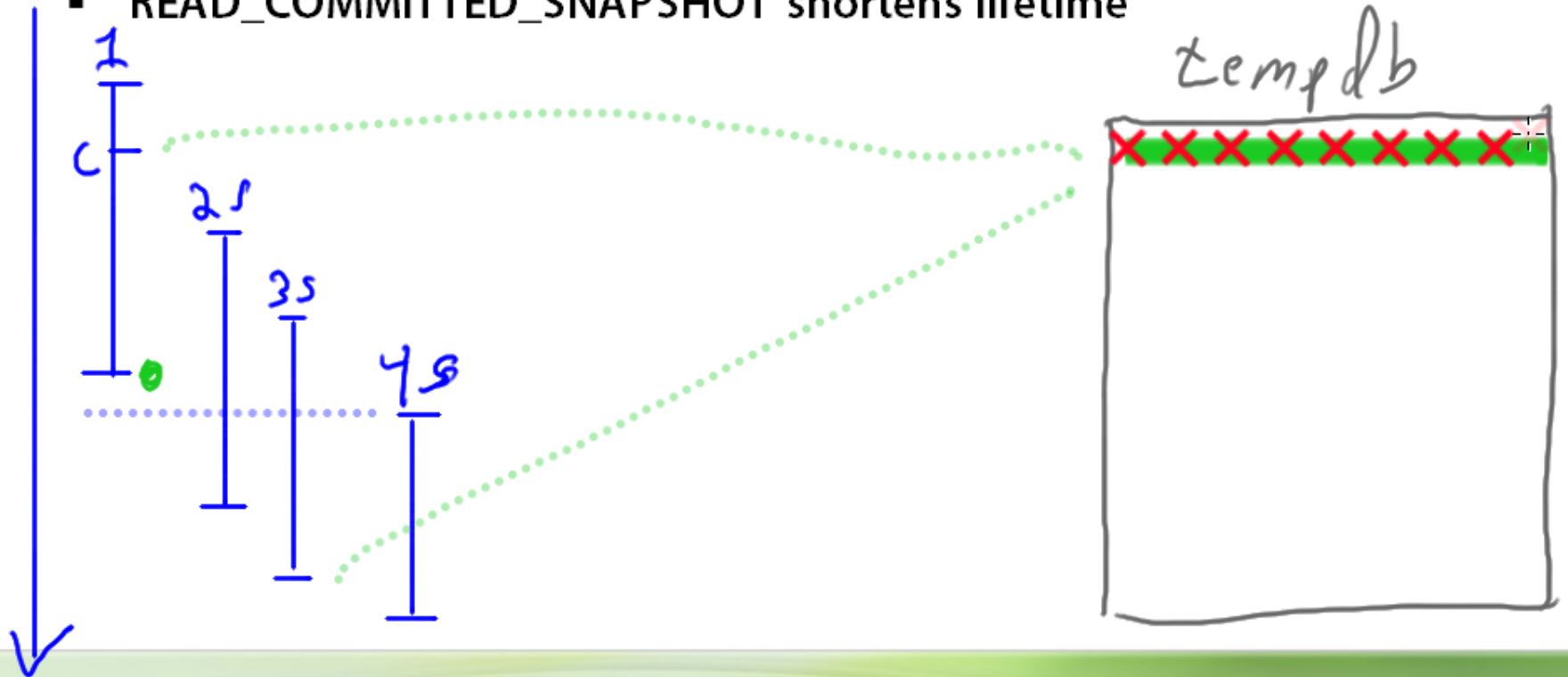


# Version Lifetime

- Simple answer
  - *long enough!*
- Until last transaction that could use it finishes
- READ\_COMMITTED\_SNAPSHOT shortens lifetime

# Version Lifetime

- Simple answer
  - *long enough!*
- Until last transaction that could use it finishes
- READ\_COMMITTED\_SNAPSHOT shortens lifetime



# Summary

- Isolation used to allow simultaneous transaction to act independently
- Isolation can be implemented using versioning
- Snapshot == Serializable
- read\_committed\_snapshot == read committed using versioning
- Better? Depends on data useage, YMMV



# References

- **Principles of Transaction Processing**
  - Bernstein & Newcomer
  - <http://link.pluralsight.com/ptp>
- **Transaction Processing: Concepts & Techniques**
  - Gray & Reuter
  - <http://link.pluralsight.com/tp>